

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Trabajo Fin de Grado

Diseño de Precodificadores para Sistemas MIMO Masivos

Autor: Ainhoa Velacoracho Velasco
Grado en Ingeniería en Tecnologías de Telecomunicación

Tutor: Matilde Pilar Sánchez Fernández

Leganés, 2015

Índice

Introduction	22
1.1. ¿What is MIMO?	22
1.1.1. Massive MIMO system.	24
1.1.2. Massive MIMO system's capacity.	24
1.2. Motivation and objectives	25
1.3. Structure.....	26
Introducción	27
2.1. ¿Qué es MIMO?	27
2.1.1 Sistemas MIMO masivo.....	29
2.1.2. Capacidad en un sistema MIMO masivo.....	29
2.2. Motivación y objetivos	30
2.3. Estructura	31
Canal H	33
Escenarios.....	36
4.1. Precodificador y filtro receptor.....	36
4.2. Filtro adaptado	39
4.3. Filtro adaptado con error	39
4.4. Filtro adaptado con limitación de bits	40
Resultados	41
5.1. Enlace ascendente.....	41
5.2. Enlace descendente	46
5.3. Otros estudios realizados	49
Líneas futuras	52
Plan del proyecto	54
7.1. Planificación	54
7.2. Presupuesto	55
Conclusions	56
Conclusiones	57
Código Matlab.....	58
Canal_jj.m.....	58

posicion_tx_rx.m	59
CanalH.m	61
MatchedFilter.m.....	62
MatchedFilterError.m	63
SampleFilter3bit.m.....	64
SampleFilter5bit.m.....	65
SampleFilter10bit.m.....	66
Compare.m.....	67
Ascendente.m	70
Descendente.m	73
MatchedFilterErrorCompare.m.....	76
SampleFilter10bitCompare.m	77

Lista de figuras

Figura 1. MIMO system.	7
Figura 2. MIMO links.	8
Figura 3. MIMO system capacity.	8
Figura 4. Precoder scheme.	11
Figura 5. Results.	12
Figura 6. Sistema MIMO.	14
Figura 7. Enlaces del canal.	15
Figura 8. Capacidad de un canal MIMO.	16
Figura 9. Esquema del precodificador.	18
Figura 10. Resultados.	19
Figura 11. Scheme of a MIMO system.	23
Figura 12. Esquema del canal de un sistema MIMO.	28
Figura 13. Esquema del precodificador en un sistema MIMO en el canal ascendente.	37
Figura 14. Esquema del filtro receptor en un sistema MIMO en el canal descendente.	38
Figura 15. Comparación de los 3 filtros.	42
Figura 16. Capacidad del escenario con 10 antenas en transmisión y 2 en recepción.	43
Figura 17. Capacidad del escenario con 20 antenas en transmisión y 1 en recepción.	43
Figura 18. Capacidad del escenario con 20 antenas en transmisión y 2 en recepción.	44
Figura 19. Capacidad del escenario con 40 antenas en transmisión y 1 en recepción.	44
Figura 20. Capacidad del escenario con 40 antenas en transmisión y 2 en recepción.	45
Figura 21. Capacidad del escenario con 64 antenas en transmisión y 40 en recepción.	45
Figura 22. Capacidad del escenario con 1 antena en transmisión y 40 en recepción.	46
Figura 23. Capacidad del escenario con 2 antenas en transmisión y 10 en recepción.	47
Figura 24. Capacidad del escenario con 1 antena en transmisión y 20 en recepción.	47
Figura 25. Capacidad del escenario con 2 antenas en transmisión y 20 en recepción.	48
Figura 26. Capacidad del escenario con 2 antenas en transmisión y 40 en recepción.	48
Figura 27. Capacidad del escenario con 40 antenas en transmisión y 64 en recepción.	49
Figura 28. Capacidad del filtro adaptado con error para valores de sigma entre 0 y 0.001.	50
Figura 29. Capacidad del filtro adaptado con error para valores de sigma entre 0 y 1.	51

Lista de fórmulas

Fórmula 1. Channel matrix H .	9
Fórmula 2. Capacity of the uplink.	10
Fórmula 3. Channel model.	10
Fórmula 4. Matriz de canal H .	16
Fórmula 5. Capacidad del enlace ascendente.	17
Fórmula 6. Modelo de canal.	18
Fórmula 7. Capacity of uplink.	24
Fórmula 8. Capacity of downlink.	25
Fórmula 9. Trace of matrix W .	25
Fórmula 10. Capacidad en el enlace ascendente.	29
Fórmula 11. Capacidad en el enlace descendente.	30
Fórmula 12. Traza de la matriz W .	30
Fórmula 13. Matriz de canal.	33
Fórmula 14. Elementos de la matriz de canal.	33
Fórmula 15. Matriz de canal desglosada.	34
Fórmula 16. Matriz de canal compacta.	34
Fórmula 17. Matriz de dispersión.	34
Fórmula 18. Matriz de canal usada.	35
Fórmula 19. Fórmula genérica del precodificador.	36
Fórmula 20. Fórmula desglosada del precodificador.	36
Fórmula 21. Fórmula del modelo general para el enlace ascendente.	37
Fórmula 22. Fórmula final desglosada.	37
Fórmula 23. Filtro adaptado.	39
Fórmula 24. Filtro adaptado con error.	39
Fórmula 25. Filtro limitado en bits.	40
Fórmula 26. Filtro adaptado con error y posteriormente limitado en bits.	52
Fórmula 27. Matriz de autocorrelación de la señal de información en CSI o CDI.	53
Fórmula 28. Matriz H en CSI o CDI.	53

Abstract

Nowadays, wireless networks have been taking over from wired ones, becoming the greatest form of communication used today.

Whenever the speed demanded by users to communicate is greater, wireless systems need to move very quickly to offer these features to people.

Various forms of communication have been established with wireless networks, being MIMO, the system we will focus during the project, one of the newest.

A MIMO system puts multiple antennas in both the transmitter side and the receiver, taking advantage of space resources, and allowing the system speed to be increased because the larger number of transmitting and receiving elements signals.

A graphical way to see it would be:



Figura 1. MIMO system.

More specifically, in this work we will study the massive MIMO systems.

The difference between normal MIMO and massive one is that the number of antennas being used is going to be bigger in massive one. While in normal MIMO is about 10 antennas, in massive MIMO it could be even 100.

In order to analyse the characteristics of massive MIMO system and check if its use in real life is feasible, the idea for this project was born.

This study presented below is based on determining the channel capacity of a massive MIMO system by placing 3 different filters in the transmitter side.

Since the study will examine both the uplink channel (which runs from the user terminal to the base station) and downlink (just the opposite), the transmitter side will change from one side to the other, being in the first case the user terminal and the BTS in the second one.

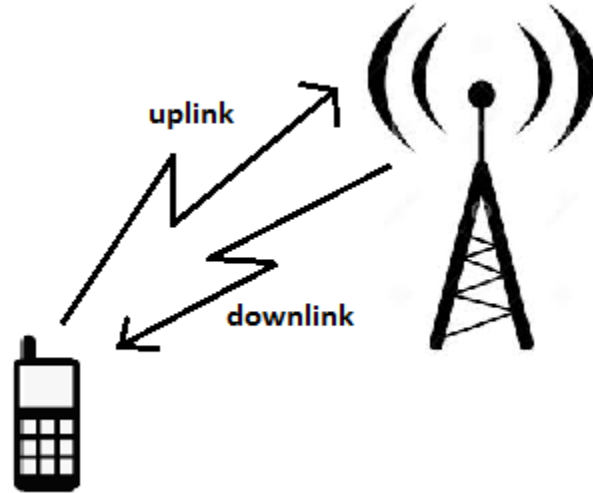


Figura 2. MIMO links.

However, the study will be always done in the user terminal side. Throughout the project, we will explain how this affects to the results.

The feature that will be studied in the project is the channel capacity. The formulas and the characteristics of the matrix which defines it will be detailed later.

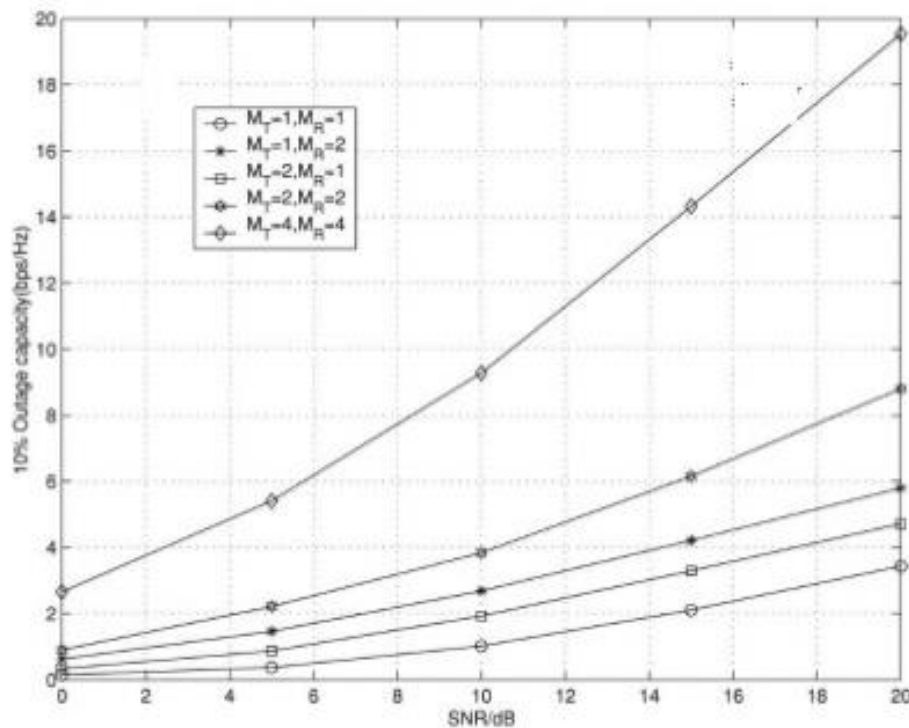


Figura 3. MIMO system capacity.

In picture 3, we can see the tendency of the capacity of a MIMO system without any filter. This way, we can compare our results with this graphic.

Logically, the bigger number of antennas used, the bigger the capacity.

Now, we are going to expose the main features of the channel.

It is with a stationary flat fading channel.

The characteristics of the MIMO channel can be defined through its channel matrix $\bar{\bar{H}}$.

$$\bar{\bar{H}} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & \ddots & & H_{2M} \\ \vdots & & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{pmatrix}$$

Fórmula 1. Channel matrix H.

This matrix has to have NxM dimensions, where M is the number of antennas in the transmitter side and N in the receiver side.

Every element of the matrix $H_{N-thM-th}$ defines the relation between the signal transmitted by the emisor and the signal received by the receiver.

$\bar{\bar{H}}$ matrix is being formed from some other matrix. Some of their characteristics are:

- Matrix which follows a Gaussian distribution whose elements are also Gaussian, independent and not identically distributed.
- Complex random matrix whose components are independent and identically distributed and follow a Gaussian distribution with an unitary variance.
- Deterministic standard diagonal matrices whose diagonal is determined by the power of the transmitter and receiver.

As we have said before, this project is going to focus on the study of the channel capacity.

The following formula defines the capacity of a MIMO system channel in the uplink:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{M} * \bar{H} * \bar{W} * \bar{R}_u * \bar{W}^H * \bar{H}^H \right| \right)$$

Fórmula 2. Capacity of the uplink.

Being:

- \bar{I} : identity matrix on NxN dimensions.
- SNR: signal-noise relation.
- M: number of antennas in the transmitter side.
- N: number of antennas in the receiver side.
- \bar{H} : channel matrix (\bar{H}^H is the hermitic one).
- \bar{W} : filter matrix (\bar{W}^H is the hermitic one).
- \bar{R}_u : transmitter autocorrelation matrix.

Once we have explain the channel model, we will focus on the precoder.

The formula of the general model of the uplink is:

$$\bar{y} = \bar{H} * \bar{x} + \bar{n}$$

Fórmula 3. Channel model.

Where \bar{H} is the matrix of the channel, with NxM dimensions, \bar{y} is the received signal, with Nx1 dimensions, \bar{x} is the transmitted signal, with Mx1 dimensions and \bar{n} is the withe and gaussian noise of Nx1 dimensions.

To include the filter in the precoder, we are going to divide the \bar{x} vector into two submatrix of MxN and Nx1 dimensions respectively, \bar{W} (the filter) and \bar{u} (the signal with the information).

The way that the precoder works is presented now:

To begin, we send a string of bits to the receiver, which captures the channel matrix, makes its estimate and sends it to the transmitter. From there, the communication continues normally.

We can say, because of this, that there exists feedback in the system. But this only happens in the uplink (we will explain it later).

This picture shows with more details how and where the feedback happens in both links.

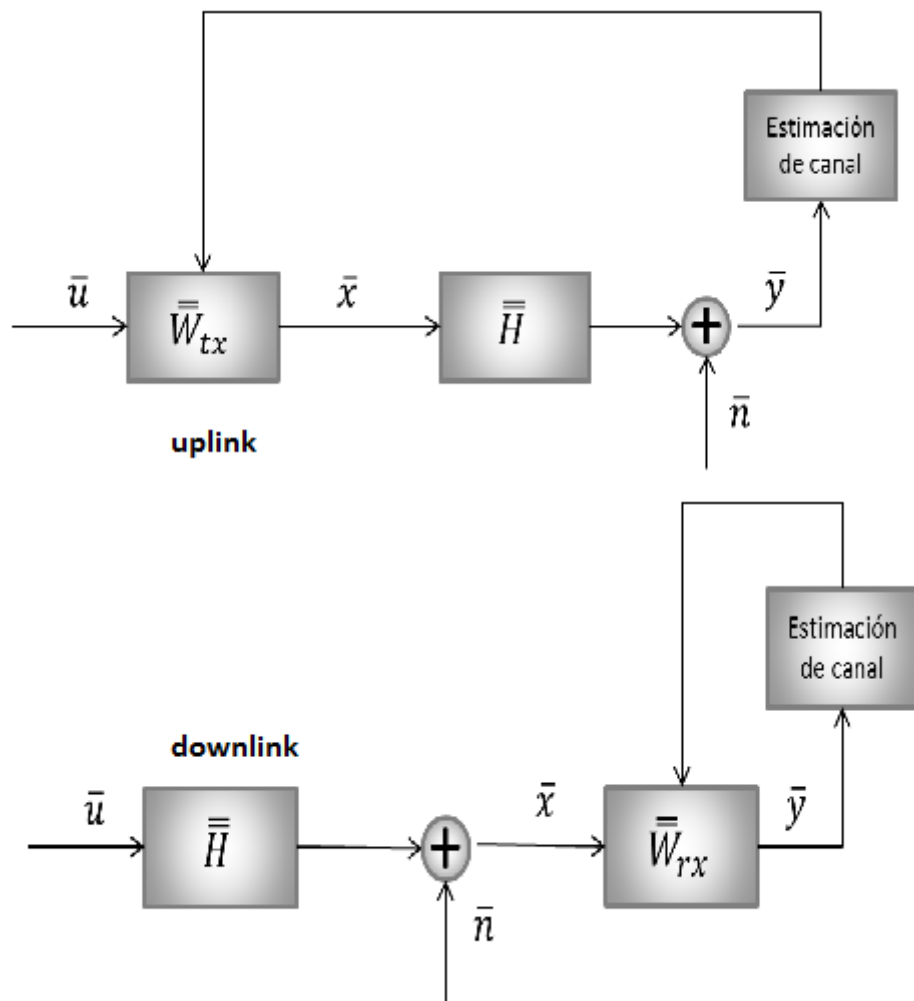


Figura 4. Precoder scheme.

As we have said before, we are going to study the precoder by using 3 types of filter.

- Matched filter: we transpose the channel matrix.
- Matched filter with error: the same as before, but in this case, we include an estimation error.
- Limited bit filter: it is a matched filter too, but in this case, we have a finite number of bits to do the estimation.

Once we have implemented these filters on Matlab, we obtain the following results:

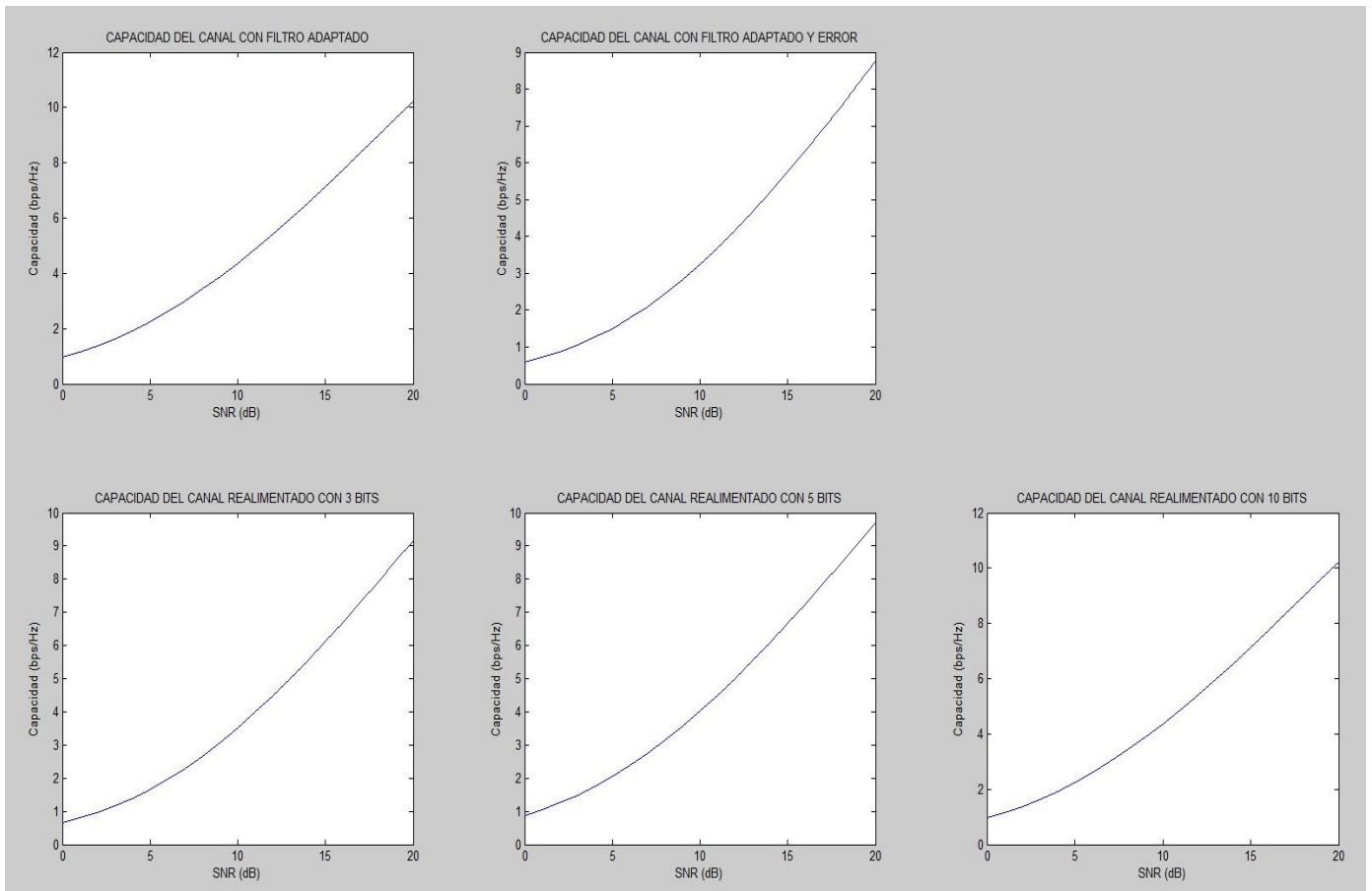


Figura 5. Results.

In this picture, which is the capacity of the uplink with the different filters, we can see that the filter that offers the best capacity is the matched one, followed by the limited to 10 bits one, and finishing with the filter with an error of estimation.

Later, we will see that also the greater the number of antennas used the higher capacity.

In the downlink, as we are going to see later, the capacity of the channel is smaller than the one from the uplink, because the influence of the noise (which is not attenuated).

Also we are going to stand out the behaviour of the limited filter in the downlink, which give us the same capacity as the matched one, because in the downlink there is no feedback.

With all these, we conclude that MIMO system is a good idea to improve the speed of the communication, which is the reason we started this project.

Resumen

En los últimos tiempos, las redes inalámbricas han ido tomando el relevo a las redes cableadas, hasta convertirse en la mayor forma de comunicación usada hoy en día.

Cada vez es mayor la velocidad que demandan los usuarios para poder comunicarse, y los sistemas inalámbricos tienen que avanzar de manera muy rápida para poder ofrecer esas características a las personas.

Diversas formas de comunicación han sido establecidas por medio de redes inalámbricas, siendo MIMO, el sistema en el que nos centraremos durante el proyecto, una de las más novedosas.

Un sistema MIMO consiste en la colocación de varias antenas tanto en el lado del transmisor como en el receptor, aprovechando de esta manera mejor los recursos espaciales, y permitiendo que la velocidad del sistema sea mayor al tener un mayor número de elementos transmitiendo y recibiendo las señales.

Una manera gráfica de verlo sería la siguiente:



Figura 6. Sistema MIMO.

Más concretamente, en este trabajo vamos a estudiar los sistemas MIMO masivos.

La diferencia con un sistema MIMO corriente es que el número de antenas que vamos a colocar es bastante grande. Mientras que un sistema MIMO normal gira en torno a la colocación de unas 10 antenas, en MIMO masivo podemos llegar a tener hasta 100 antenas.

Con el fin de analizar las características del sistema MIMO masivo y comprobar que su uso en la vida real es factible, nace la idea de realizar este proyecto.

Este estudio que presentamos a continuación se basa en determinar la capacidad del canal de un sistema MIMO masivo mediante la colocación de 3 diferentes filtros en el lado del transmisor.

Puesto que el estudio va a estudiar tanto el enlace ascendente del canal (el que va desde el terminal de usuario hasta la estación base) como el descendente (justo al revés), el transmisor cambiará de un escenario a otro, siendo en el primer caso el terminal y en el segundo la BTS.

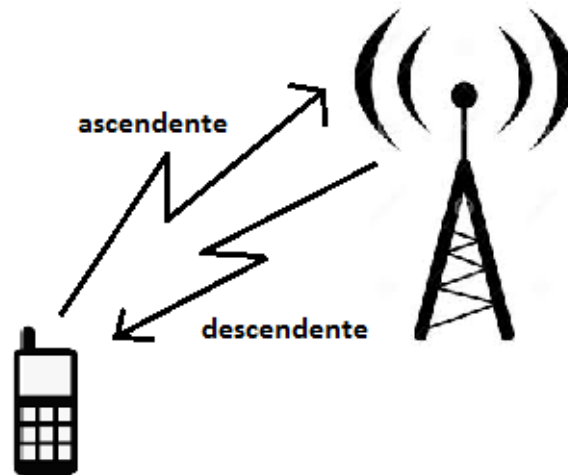


Figura 7. Enlaces del canal.

Sin embargo, el estudio siempre va a realizarse desde el lado del canal en el que se encuentra el terminal. A lo largo del proyecto iremos explicando cómo afecta esto a los resultados.

La característica que se va a estudiar en el proyecto es la capacidad del canal. Las fórmulas se detallarán más adelante así como las características de la matriz que lo define.

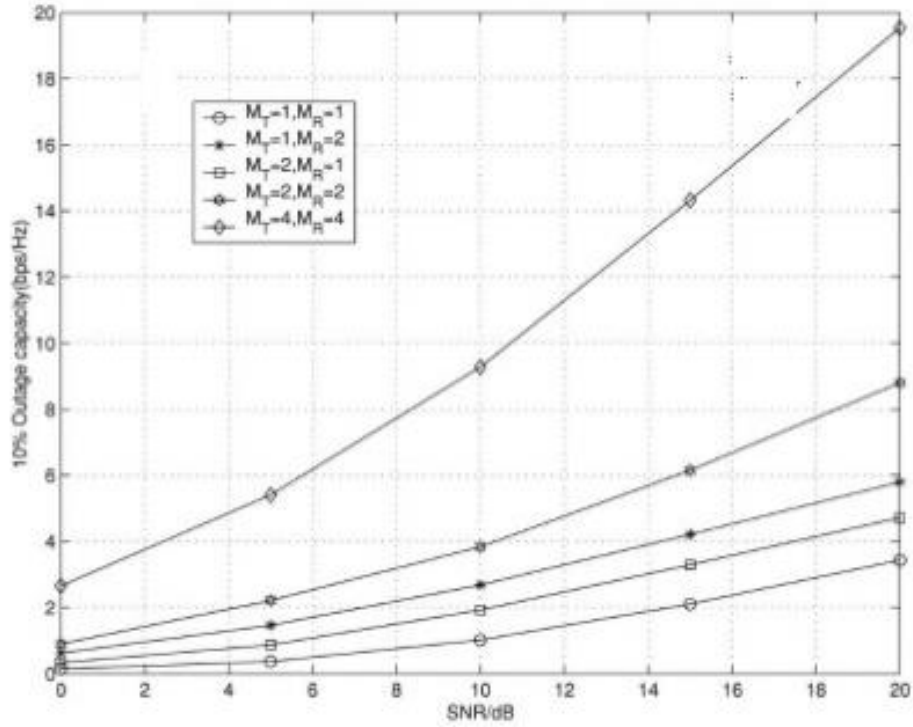


Figura 8. Capacidad de un canal MIMO.

En la figura 8, observamos la tendencia que tiene la capacidad de un sistema MIMO normal, sin filtros, para poder compararla con nuestros resultados.

Como es lógico, cuanto mayor es el número de antenas, mayor es la capacidad obtenida.

A continuación, vamos a exponer las principales características del canal del sistema.

Se trata de un canal estacionario y con desvanecimiento plano.

Las características del canal MIMO se pueden definir a través de su matriz de canal $\bar{\bar{H}}$.

$$\bar{\bar{H}} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & \ddots & & H_{2M} \\ \vdots & & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{pmatrix}$$

Fórmula 4. Matriz de canal H.

Esta matriz ha de ser de dimensiones NxM, donde M es el número de antenas en transmisión y N en recepción.

Cada elemento de la matriz $H_{N-\acute{e}simoM-\acute{e}simo}$ define la relación entre la señal que transmite el emisor y la que recibe el receptor.

La matriz $\bar{\bar{H}}$ ha sido formada a partir de varias matrices, algunas de las cuales tienen características tales como:

- Matriz que siguen una distribución gaussiana cuyos elementos también son gaussianos independientes y no idénticamente distribuidos.
- Matriz aleatoria compleja cuyos componentes son independientes y no idénticamente distribuidos y siguen una distribución gaussiana de varianza unidad.
- Matrices diagonales deterministas normalizadas cuya diagonal viene determinada por la potencia del transmisor y del receptor deterministas que dependerán únicamente de la geometría de la antena.

Como decíamos al principio, este trabajo se va a centrar en el estudio de la capacidad del canal.

La siguiente fórmula define la capacidad del canal en un sistema MIMO para el enlace ascendente:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{M} * \bar{H} * \bar{W} * \bar{R}_u * \bar{W}^H * \bar{H}^H \right| \right)$$

Fórmula 5. Capacidad del enlace ascendente.

Siendo:

- \bar{I} : la matriz identidad de dimensiones $N \times N$.
- SNR: la relación señal-ruido.
- M: el número de antenas en transmisión.
- N: el número de antenas en recepción
- \bar{H} : la matriz del canal (\bar{H}^H es la hermítica).
- \bar{W} : la matriz del precodificador o filtro receptor a utilizar (\bar{W}^H es la hermítica).
- \bar{R}_u : la matriz de autocorrelación de transmisión.

Una vez explicado el modelo de canal, vamos a centrarnos en el precodificador.

La fórmula del modelo general del enlace es:

$$\bar{y} = \bar{H} * \bar{x} + \bar{n}$$

Fórmula 6. Modelo de canal.

Donde \bar{H} es la matriz del canal, de dimensiones NxM, \bar{y} es la señal recibida, de dimensiones Nx1, \bar{x} es la señal transmitida, de dimensiones Mx1 y \bar{n} es el ruido gaussiano y blanco de dimensiones Nx1.

Para incluir el filtro en el precodificador, vamos a dividir el vector \bar{x} en dos submatrices de dimensiones MxN y Nx1 respectivamente, \bar{W} (el filtro a utilizar) y \bar{u} (la señal con la información).

El funcionamiento del precodificador es el siguiente:

Para empezar, mandamos un conjunto de bits al receptor, el cual capta la matriz de canal, realiza su estimación y se la envía al transmisor. A partir de ahí comienza la transmisión normal de datos.

Podemos decir, por tanto, que existe una realimentación en el sistema. Pero este fenómeno sólo ocurre en el enlace ascendente (explicamos el por qué más adelante).

En la siguiente figura podemos apreciar con más detalle donde ocurre la realimentación según el enlace en el que nos encontramos.

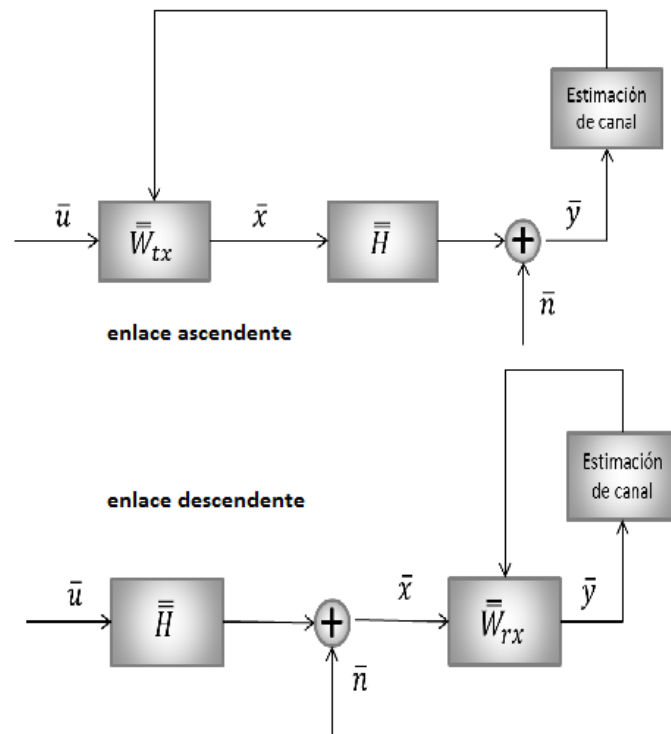


Figura 9. Esquema del precodificador.

Como hemos explicado antes, vamos a estudiar el precodificador a partir de 3 tipos de filtros.

- Filtro adaptado: se consigue trasponiendo la matriz de canal.
- Filtro adaptado con error: el mismo de antes, pero incluyéndole un error de estimación.
- Filtro limitado en bits: también está adaptado, pero en este caso, contamos con un número finito de bits para realizar la estimación.

Una vez implementados estos filtros en Matlab, obtenemos los siguientes resultados:

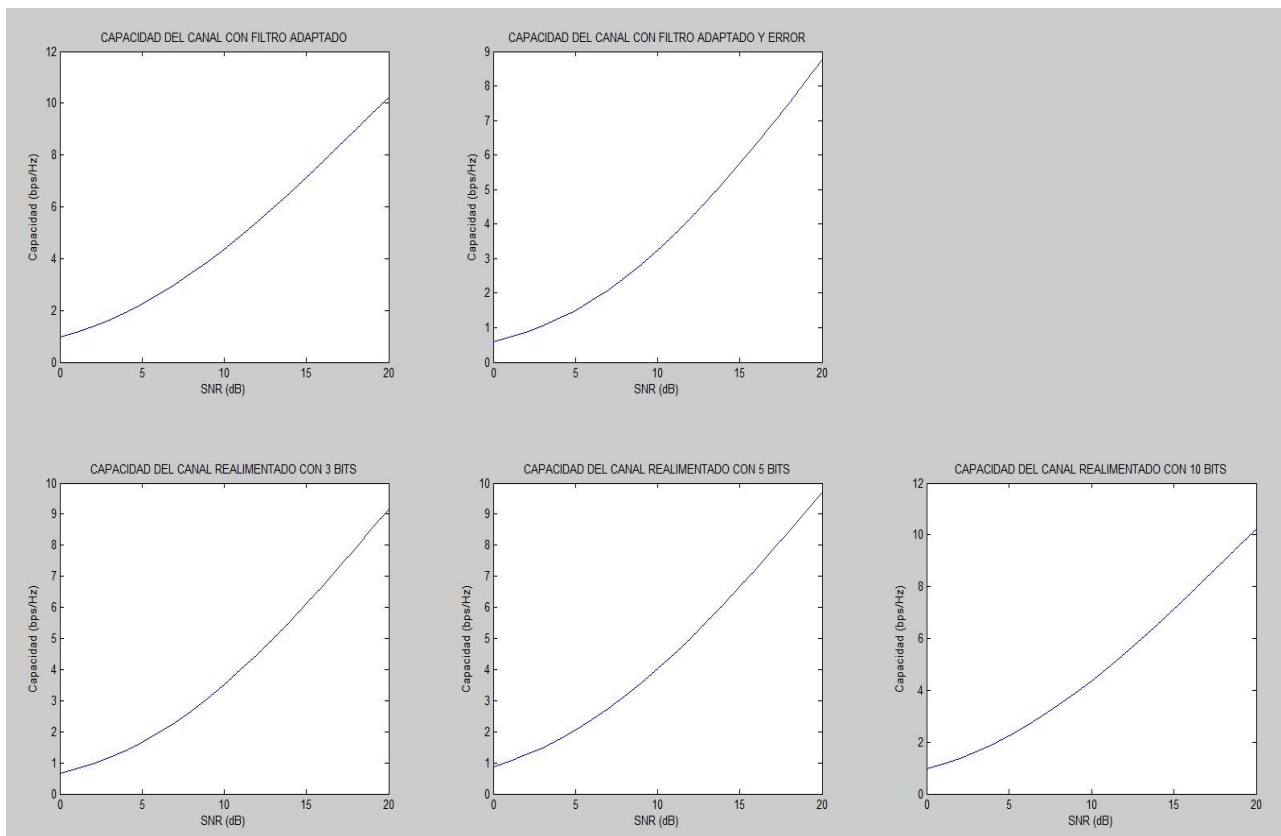


Figura 10. Resultados.

En la figura 5, correspondiente al enlace ascendente, observamos que el filtro que mayor capacidad nos ofrece es el filtro adaptado, seguido del limitado a 10 bits, y terminando por el que contenía un error.

En figuras posteriores podremos observar que también aumenta la capacidad cuanto mayor es el número de antenas empleadas.

Por otro lado, en el enlace descendente, como veremos más tarde, la capacidad máxima disminuye debido a la influencia del ruido que no es atenuado. Además, destacaremos el caso del filtro limitado en bits, ya que al no haber realimentación, nos ofrece la misma capacidad que el filtro adaptado.

Con todo esto, llegamos a la conclusión final de que, efectivamente, el uso de sistemas MIMO masivos es factible, permitiendo una mayor velocidad de transmisión, que es lo que buscábamos desde el principio, satisfacer las necesidades de los consumidores.

Capítulo 1

Introduction

Now, we are going to introduce you to MIMO systems, specially the massive ones, as well as their capacity.

We also include a section with the reasons that took us to undertake this project, with the objectives that we pursue, and a plan about how the document is structured.

1.1. ¿What is MIMO?

In the last years, wireless networks have become the most important form of communication for the entire population. Several structures have been established, from infrared systems to 5G networks [1].

The demand from users to increase transmission rate and to reduce errors in the channel, leads to the conclusion that an increase of the number of antennas transmitting and receiving data in wireless systems is required. That is why MIMO (multiple-input multiple-output) systems starts.

A MIMO system is a technology that is based on the transmission of information over a channel by using more than one antenna at the transmitter and receiver, as its name suggests.

By increasing the number of antennas used in the transmission we pretend to increase the capacity of the channel and reduce the probability of error. It means achieve a greater spectral efficiency in the system through a better use of space resources using more antennas [2].

This is easy to explain. On the one hand, if we divide the quantity of information into some antennas, we obtain a bigger capacity of the channel. On the other hand, if all antennas transmit the same information, no matter if some antennas fail, because the information arrives good to the other extreme through the other antennas. That means a smaller probability of error in the channel.

The larger the number of antennas is the more freedom we have to improve the channel performance.

This graphic shows the scheme of a MIMO system [3].

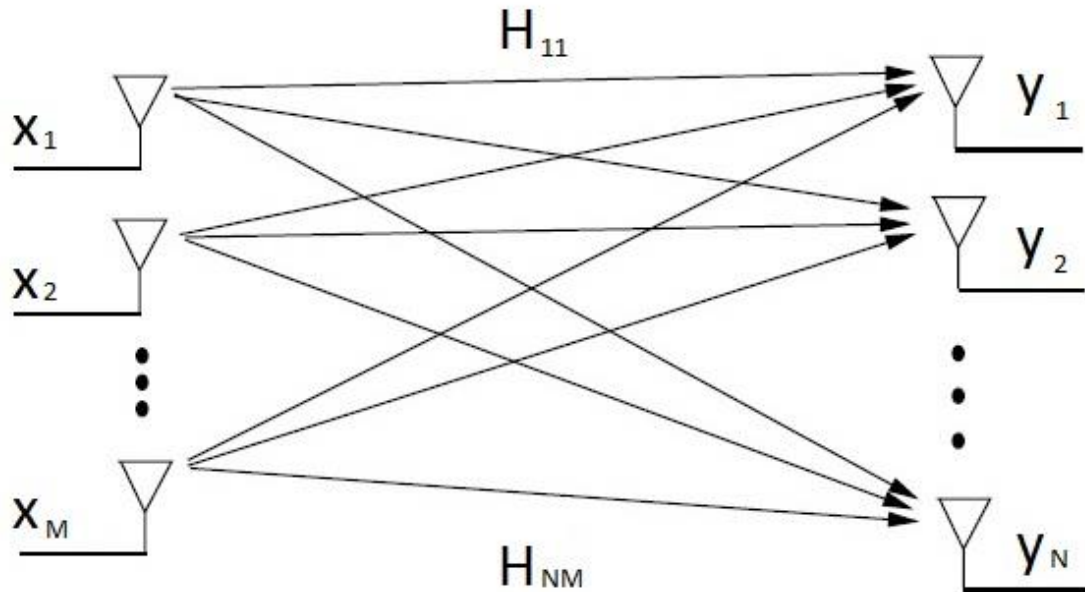


Figura 11. Scheme of a MIMO system.

Now, we are going to describe some different technologies that uses MIMO system:

- Fourth generation mobile networks (4G) combines MIMO system with OFDM modulation, and is present in technologies such as LTE and WiMAX [4].
- It is expected to be part of the future mobile 5G networks [1].
- It is included in IEEE 802.11n standard as one system involved in wireless local area network (WLAN) [5].
- Another technologies like RFID, which is used, for example, in the idea of IoT (Internet of Things) [6].

1.1.1. Massive MIMO system.

In a massive MIMO system (also named very large MIMO) the number of antennas is bigger than 100, while conventional MIMO would use a smaller number, in some systems for example the minimum number is 8 antennas. These antennas serve simultaneously a small number of terminals [7].

The antennas used in this technology use low power on the order of milliwatts, but maintain constant the total power.

One advantage of using a large number of antennas is that the system is less sensitive to a failure of any of them.

The main problem with these systems is the physical space, which limits the maximum number of possible antennas to be used.

1.1.2. Massive MIMO system's capacity.

In this project, we are going to study the massive MIMO system when using different precoders and/or receive filters both in uplink and downlink.

It should be noted then we are always placing the signal processing of the MIMO system that compensates the channel in the user side. This is not the conventional way of doing things, however we thought this way any number of antennas can be deployed in the user side, without affecting the network and therefore the BTS. That's the reason why we refer to uplink the communication link that goes from the user terminal to the base station. In this case we put the precoder on the transmitter side.

The downlink is the one that goes from the base station to the user terminal. In this case we put the filter on the receiver side.

On the one hand, the channel capacity in the uplink is determined by the following formula [3]:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{M} * \bar{H} * \bar{W} * \bar{R}_u * \bar{W}^H * \bar{H}^H \right| \right)$$

Fórmula 7. Capacity of uplink.

Where I is the identity matrix of $N \times N$ dimensions, SNR is the relation between the signal and the noise, M is the number of antennas in transmission, N is the number of antennas in reception, \bar{H} is the channel matrix (\bar{H}^H is the hermitic one), \bar{W} is the precoder matrix (\bar{W}^H is the hermitic one) and \bar{R}_u is the autocorrelation matrix of transmission (during the project we consider that is the identity matrix).

On the other hand, the channel capacity in the downlink is determined by the following formula [3]:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{N} * \bar{W} * \bar{H} * \bar{R}_u * \bar{H}^H * \bar{W}^H * (\bar{W} * \bar{W}^H)^{-1} \right| \right)$$

Fórmula 8. Capacity of downlink.

Where $()^{-1}$ is the inverse matrix of the product $\bar{W} * \bar{W}^H$. In this case, we had to invert dimensions of \bar{W} (now $N \times M$) and \bar{H} (now $M \times N$), because now the emisor is the BTS, and that's the reason why \bar{I} has $M \times M$ dimensions now too.

It is necessary to divide by M and N respectively in formula 1 and 2 so as to the total transmitted power is the same, because as we can see in formula 3, we force the trace of the matrix filter to M in uplink (in downlink is force to N), and therefore it is necessary to divide by M (or N) then to normalize the power.

$$tr \{ \bar{W} * \bar{W}^H \} = M$$

Fórmula 9. Trace of matrix W.

1.2. Motivation and objectives

As we have seen before, a massive MIMO system involves placing a big number of antennas both in transmission and reception so as to increase the capacity of the channel.

With the objective of discover if the idea of massive MIMO is right, we decide to make this project, in which we are going to study different types of scenarios that can be done in a real system by simulating the performance of the different models of precoder for both uplink and downlink, specially by the study of the of the channel.

This project continues the study of a former student of the university [8]. In her future lines, she propose to study different types of precoders for MIMO system. Also, we use some Irene's Matlab code to do my project.

1.3. Structure

This bachelor thesis is divided into 9 main chapters.

- Chapter 1: English introduction to massive MIMO system. Also we include the motivation and objectives of the project and an explanation of the structure of the document.
- Chapter 2: The spanish version of chapter 1.
- Chapter 3: Here we explain with many details the features of the channel matrix $\bar{\bar{H}}$.
- Chapter 4: A little introduction about what is a precoder and the presentation of the 3 scenarios that we are going to study, including their formulas and how we expect they are going to work.
- Chapter 5: The simulation of the results.
- Chapter 6: We propose in this chapter some new lines of study related to this topic.
- Chapter 7: It includes the way that this project has been made and the budget needed to develop it.
- Chapter 8: Final conclusions of the Project (in english).
- Chapter 9: the spanish version of chapter 8.

Furthermore, there are other important sections, like the general index, the index of pictures and formulas, the summary (both in Spanish and English), the annex with the Matlab code and the bibliography used.

Capítulo 2

Introducción

A continuación, haremos una introducción sobre los sistemas MIMO, más concretamente los masivos, y sobre su capacidad.

Además, incluimos un apartado sobre la motivación que nos ha llevado a realizar este proyecto, sobre los objetivos que perseguimos, y un esquema sobre cómo está estructurado el documento.

2.1. ¿Qué es MIMO?

Durante los últimos años, las redes inalámbricas han pasado a ser la forma de comunicación más importante para toda la población. Diversas estructuras han sido establecidas, desde sistemas infrarrojos hasta redes 5G [1].

La demanda por parte de los usuarios de aumentar la velocidad de transmisión y de reducir los errores en el canal, finalmente nos lleva a la conclusión de que es necesario un aumento del número de antenas en la transmisión y recepción de datos en los sistemas inalámbricos. Es por ello que nacen los sistemas MIMO (multiple-input multiple-output).

Un sistema MIMO es una tecnología que se basa en la transmisión de información por un canal mediante el uso de más de una antena en el transmisor y en el receptor, como su propio nombre indica.

La siguiente gráfica muestra el esquema de un sistema MIMO [3].

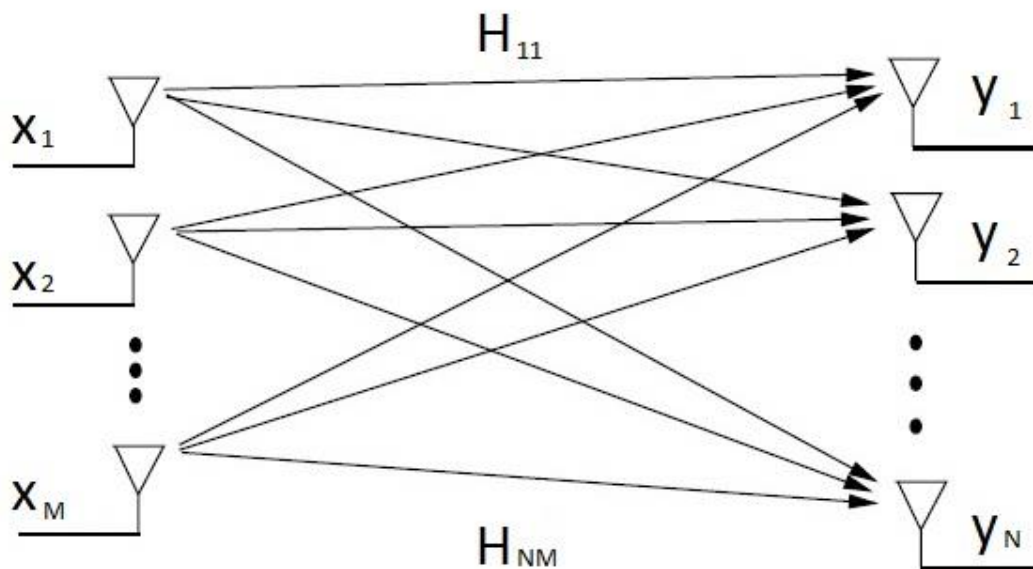


Figura 12. Esquema del canal de un sistema MIMO.

Mediante el aumento del número de antenas a usar en la transmisión se pretende aumentar la capacidad del canal y reducir la probabilidad de error, es decir, conseguir una mayor eficiencia espectral en el sistema mediante un mejor uso de los recursos espaciales al utilizar un mayor número de antenas [2].

La explicación de ese fenómeno es sencilla. Por un lado, si las antenas se reparten la cantidad de información a transmitir, aumenta la capacidad del canal. Por otro lado, si todas las antenas transmiten la misma información, es más probable que, aunque falle alguna, la información llegue correctamente al otro extremo, con lo que se reduce la probabilidad de error del canal.

Cuanto mayor es el número de antenas usado en el sistema, mayor grado de libertad tenemos para mejorar las prestaciones del canal.

Cabe destacar que los sistemas MIMO están siendo ya incorporados a diferentes tecnologías que enumeramos a continuación:

- Las redes móviles de cuarta generación (4G): Combina el sistema MIMO con una modulación OFDM, y se encuentra presente en tecnologías tales como LTE y WiMAX [4].
- Está prevista su incorporación a las redes móviles 5G del futuro [1].
- Las redes inalámbricas (WLAN) lo recogen como sistema a utilizar en el estándar 802.11n del IEEE [5].
- Otras tecnologías como los sistemas multiantena en la tecnología RFID, utilizados, por ejemplo para el concepto de IoT (Internet of Things) [6].

2.1.1 Sistemas MIMO masivo.

En un sistema MIMO masivo (very large MIMO en inglés) el número de antenas del que hablamos es superior a 100, mientras que en MIMO convencional el número de antenas utilizadas es muy bajo, en algunos sistemas pueden llegar a 8. Estas antenas sirven simultáneamente a un número reducido de terminales [7].

Las antenas utilizadas en esta tecnología usan potencias bajas, del orden de milivatios, pero mantienen la potencia transmitida total constante según aumenta su número en la BTS.

Una de las ventajas del uso de un número elevado de antenas es que el sistema es menos sensible al fallo de alguna de ellas.

El principal problema de estos sistemas es el espacio físico, que limita el máximo de posibles antenas a ser usadas.

2.1.2. Capacidad en un sistema MIMO masivo.

En este proyecto, vamos a realizar un estudio de la capacidad del canal en función de diferentes precodificadores y filtros receptores tanto en el enlace ascendente como en el descendente.

Cabe destacar que estamos realizando el procesamiento de señal siempre en el lado del terminal de usuario. Esta no es la forma tradicional de trabajar, pero pensamos que es la mejor opción ya que así tenemos total libertad para colocar cualquier número de antenas en el lado del usuario sin afectar ni a la red ni a la BTS.

Por ello, nos referimos a enlace ascendente cuando hablamos del canal que va del terminal a la estación base. En este caso ponemos el precodificador en el lado del transmisor.

Por otro lado, el enlace descendente es el que va de la estación base al terminal. En él, situamos el filtro en el lado del receptor.

La capacidad del canal en el enlace ascendente viene determinada por la siguiente fórmula [3]:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{M} * \bar{H} * \bar{W} * \bar{R}_u * \bar{W}^H * \bar{H}^H \right| \right)$$

Fórmula 10. Capacidad en el enlace ascendente.

Donde \bar{I} es la matriz identidad de dimensiones $N \times N$, la SNR es la relación señal-ruido expresada en dB, M es el número de antenas en transmisión, N es el número de antenas en recepción, \bar{H} es la matriz del canal (\bar{H}^H es la hermítica), \bar{W} es la matriz del precodificador o filtro receptor a utilizar (\bar{W}^H es la hermítica) y \bar{R}_u es la matriz de autocorrelación de transmisión, aunque durante todo el proyecto nos olvidaremos de ella ya que valdrá a identidad.

Por otro lado, la capacidad del canal en el enlace descendente viene dada por [3]:

$$R = \log_2 \left(\left| \bar{I} + \frac{SNR}{N} * \bar{W} * \bar{H} * \bar{R}_u * \bar{H}^H * \bar{W}^H * (\bar{W} * \bar{W}^H)^{-1} \right| \right)$$

Fórmula 11. Capacidad en el enlace descendente.

En este caso $()^{-1}$ es la inversa del producto $\bar{W} * \bar{W}^H$. Además, cabe destacar que hemos tenido que invertir las dimensiones de \bar{W} (ahora $N \times M$) y \bar{H} (ahora $M \times N$) porque el emisor ahora es la BTS. Esto hace que también cambien las dimensiones de \bar{I} que ahora es $M \times M$.

Es necesario dividir por M y N respectivamente en las fórmulas 1 y 2 para que la potencia total transmitida sea la misma, ya que como podemos ver en la fórmula 3, forzamos a la traza de la matriz del filtro a que sea igual a M en el ascendente (en el descendente lo forzamos a N), y por tanto es necesario volver a dividir por M (o N) después, para normalizar la potencia.

$$tr \{ \bar{W} * \bar{W}^H \} = M$$

Fórmula 12. Traza de la matriz \bar{W} .

2.2. Motivación y objetivos

Como ya hemos visto antes, un sistema MIMO masivo consiste en la colocación de un número elevado de antenas tanto en transmisión como en recepción para así aumentar la capacidad del canal.

Con el objetivo de comprobar si el uso de sistemas MIMO masivo es factible, nace la idea de realizar este TFG, en el que vamos a estudiar diferentes propuestas de posibles escenarios que pueden darse en un sistema como este, mediante la simulación de las prestaciones de modelos de precodificador o filtro receptor tanto para el canal ascendente como para el descendente, y más concretamente, del estudio de la del canal al transmitir y recibir las señales.

Este proyecto continúa el estudio de una antigua alumna de la universidad [8]. En las líneas futuras de su proyecto, aparece el estudio de diferentes precodificadores para sistemas MIMO. Por eso mismo, parte del código Matlab utilizado en este trabajo (los 3 primeros .m del anexo) pertenecen a su proyecto.

2.3. Estructura

Este TFG está dividido en 9 capítulos principales.

- Capítulo 1: En este apartado hacemos una pequeña introducción en inglés a los sistemas MIMO masivo, explicamos porqué hemos elegido este tema para el proyecto y detallamos la estructura del documento.
- Capítulo 2: Incluimos la correspondiente versión en español de la introducción.
- Capítulo 3: En este capítulo se explican con bastante detalle las características de la matriz de canal $\bar{\bar{H}}$.
- Capítulo 4: Aquí hacemos una breve introducción a lo que es un precodificador y presentamos los 3 escenarios que van a ser sometidos a estudio, su funcionamiento, sus fórmulas y el comportamiento que esperamos que vayan a tener.
- Capítulo 5: En este capítulo introducimos los resultados de las simulaciones en forma de gráficas.
- Capítulo 6: En esta sección se propondrán una serie de nuevas metas a llevar a cabo en futuros proyectos relacionados con el tema.
- Capítulo 7: Se detallará cómo ha sido llevado a cabo este proyecto así como el presupuesto que ha sido necesario para su realización.
- Capítulo 8: Dedicamos este capítulo a explicar las conclusiones finales a las que se ha llegado al concluir el proyecto (en inglés).
- Capítulo 9: Por último, incluimos la correspondiente versión en español de las conclusiones.

Además, hay otros apartados que son imprescindibles para seguir bien la memoria, como el índice general, de figuras y de fórmulas, el resumen (tanto en castellano como en inglés), el anexo con el código Matlab que se ha utilizado para las simulaciones y la bibliografía a través de la cual hemos podido realizar este estudio.

Capítulo 3

Canal H

Vamos a realizar una introducción al modelo de canal que utilizó Irene en su proyecto y que nos sirve como punto de partida para este trabajo.

Un canal MIMO basado en M antenas transmisoras y N receptoras está definido por la matriz de canal $\bar{\bar{H}}$ [3], cuyos elementos definen el desvanecimiento del canal existente entre la antena transmisora en el terminal y la antena receptora en la BTS (base transceiver station).

$$\bar{\bar{H}} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & \ddots & & H_{2M} \\ \vdots & & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{pmatrix}$$

Fórmula 13. Matriz de canal.

Asumimos que este canal es estacionario y con desvanecimiento plano.

Cada elemento de la matriz se puede modelar utilizando la función de Green en la posición de la antena receptora ($\bar{r}_{N-\acute{e}sima}'$) teniendo en cuenta que la señal se emite desde la antena transmisora ($\bar{r}_{M-\acute{e}sima}$).

$$H_{MN} = \int \int \bar{S}(\bar{k}', \bar{k}) * e^{-j\bar{k}\bar{r}_M} * e^{j\bar{k}'\bar{r}_N'} * d\bar{k}' * d\bar{k}$$

Fórmula 14. Elementos de la matriz de canal.

$\bar{S}(\bar{k}', \bar{k})$ es la función de dispersión del canal, y \bar{k} y \bar{k}' son los vectores de propagación de las ondas transmitidas y recibidas.

Teniendo en cuenta los NxM elementos y muestreando el espacio de propagación en un conjunto de L y L' ondas planas respectivamente, finalmente la matriz $\bar{\bar{H}}$ queda de la siguiente manera:

$$\bar{H} = \begin{pmatrix} e^{j\bar{k}_1 r_1} & e^{j\bar{k}_1 r_2} & \cdots & e^{j\bar{k}_1 r_N} \\ e^{j\bar{k}_2 r_1} & \ddots & & e^{j\bar{k}_2 r_N} \\ \vdots & & \ddots & \vdots \\ e^{j\bar{k}_L r_1} & e^{j\bar{k}_L r_2} & \cdots & e^{j\bar{k}_L r_N} \end{pmatrix} * \begin{pmatrix} S(\bar{k}'_1, \bar{k}_1) & S(\bar{k}'_1, \bar{k}_2) & \cdots & S(\bar{k}'_1, \bar{k}_L) \\ S(\bar{k}'_2, \bar{k}_1) & \ddots & & S(\bar{k}'_2, \bar{k}_L) \\ \vdots & & \ddots & \vdots \\ S(\bar{k}'_L, \bar{k}_1) & S(\bar{k}'_L, \bar{k}_2) & \cdots & S(\bar{k}'_L, \bar{k}_L) \end{pmatrix}$$

$$* \begin{pmatrix} e^{j\bar{k}_1' r_{1'}} & e^{j\bar{k}_2' r_{1'}} & \cdots & e^{j\bar{k}_L' r_{1'}} \\ e^{j\bar{k}_1' r_{2'}} & \ddots & & e^{j\bar{k}_L' r_{2'}} \\ \vdots & & \ddots & \vdots \\ e^{j\bar{k}_1' r_{M'}} & e^{j\bar{k}_2' r_{M'}} & \cdots & e^{j\bar{k}_L' r_{M'}} \end{pmatrix}$$

Fórmula 15. Matriz de canal desglosada.

Por tanto, la matriz se puede descomponer en otras tres matrices:

$$\bar{H} = \bar{B}_N^H * \bar{S} * \bar{B}_M$$

Fórmula 16. Matriz de canal compacta.

Puesto que nos encontramos en un canal estacionario, tanto \bar{B}_M como \bar{B}_N han de ser matrices rectangulares deterministas que dependerán únicamente de la geometría de la antena. Por su parte, \bar{S} será una matriz rectangular aleatoria modelada a partir de las características angulares de la dispersión existente.

Las entradas del canal siguen una distribución gaussiana y los coeficientes de la dispersión son independientes, por lo que \bar{S} está caracterizada totalmente por el perfil angular de potencia (PAS), lo que nos lleva a que los elementos de esta son gaussianos independientes y no idénticamente distribuidos.

Puesto que el PAS del receptor y del transmisor son independientes, la distribución de potencia en las filas viene dada por el PAS en el receptor y la distribución de potencia en las columnas coincide con el PAS en el transmisor. El PAS en el receptor y en el transmisor están caracterizados respectivamente por los parámetros ASA (Angular Spread of Arrival o dispersión angular de salida en el transmisor) y ASD (Angular Spread of Departure o dispersión angular de llegada en el receptor).

Teniendo en cuenta estos parámetros, la matriz \bar{S} se puede simplificar de la siguiente forma:

$$\bar{S} = \bar{\Sigma}_{ASA}^{1/2} * \bar{G} * \bar{\Sigma}_{ASD}^{1/2}$$

Fórmula 17. Matriz de dispersión.

Donde $\bar{\bar{G}}$ es una matriz aleatoria compleja cuyos componentes son independientes y no idénticamente distribuidos y siguen una distribución gaussiana de varianza unidad, $\bar{\bar{\Sigma}}_{ASA}$ y $\bar{\bar{\Sigma}}_{ASD}$ son matrices diagonales deterministas cuya diagonal viene determinada por el PAS del transmisor y del receptor. Estas matrices están normalizadas para que su traza sea 1.

Finalmente la matriz de canal $\bar{\bar{H}}$ queda:

$$\bar{\bar{H}} = \bar{\bar{B}}_N^H * \bar{\bar{\Sigma}}_{ASA}^{1/2} * \bar{\bar{G}} * \bar{\bar{\Sigma}}_{ASD}^{1/2} * \bar{\bar{B}}_M$$

Fórmula 18. Matriz de canal usada.

Capítulo 4

Escenarios

En este capítulo vamos a explicar los 3 escenarios de precodificación y filtro receptor que sometemos a estudio, así como incluir una pequeña introducción a lo que es un precodificador.

4.1. Precodificador y filtro receptor

Como hemos dicho antes, este proyecto se centrará en el estudio de las propiedades de 3 diferentes precodificadores o filtros receptores, tanto en el enlace ascendente como en el descendente.

En todos los casos, vamos a situar el precodificador siempre en el lado del terminal, pudiendo construir dispositivos ya con estos filtros, y no teniendo que hacer cambios en las BTS si el sistema funciona como esperamos.

Recordamos que al hacer las pruebas en el terminal de usuario, nos referiremos al enlace ascendente cuando hablemos de la comunicación que va desde el terminal a la BTS. Por el contrario, el enlace descendente es aquel por el que se envía la señal desde la BTS al terminal.

La fórmula genérica del modelo general para el enlace ascendente (independientemente del precodificador usado) es [3]:

$$\bar{y} = \bar{\bar{H}} * \bar{x} + \bar{n}$$

Fórmula 19. Fórmula genérica del precodificador.

Y desglosada [3]:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & \ddots & & H_{2M} \\ \vdots & & \ddots & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{pmatrix}$$

Fórmula 20. Fórmula desglosada del precodificador.

Donde $\bar{\bar{H}}$ es la matriz del canal, \bar{y} es la señal recibida, \bar{x} es la señal transmitida y \bar{n} es el ruido gaussiano y blanco.

Una vez que sometemos el precodificador a estudio, lo que hacemos es desglosar la señal de transmisión \bar{x} como la descomposición de dos señales $\bar{\bar{W}}$ y \bar{u} , siendo $\bar{\bar{W}}$ el filtro que vamos a ir modificando en cada escenario y \bar{u} la señal con información.

La fórmula final queda de la siguiente manera:

$$\bar{y} = \bar{\bar{H}} * \bar{\bar{W}} * \bar{u} + \bar{n}$$

Fórmula 21. Fórmula del modelo general para el enlace ascendente.

Y desglosada:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1M} \\ H_{21} & & \ddots & H_{2M} \\ \vdots & & & \vdots \\ H_{N1} & H_{N2} & \cdots & H_{NM} \end{pmatrix} * \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & & \ddots & W_{2N} \\ \vdots & & & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{MN} \end{pmatrix} * \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{pmatrix}$$

Fórmula 22. Fórmula final desglosada.

Vamos a explicar el funcionamiento del precodificador. Tras un primer envío de un conjunto de pilotos para la estimación de canal, el receptor capta la matriz de canal $\bar{\bar{H}}$, y hace una estimación del canal. Posteriormente, envía esta información al transmisor y continúa la comunicación. Según el número de bits que utilicemos para codificar dicha matriz de estimación, tenemos más o menos precisión.

Podemos ver un ejemplo más gráfico del modelo de precodificador y filtro receptor en las siguientes imágenes:

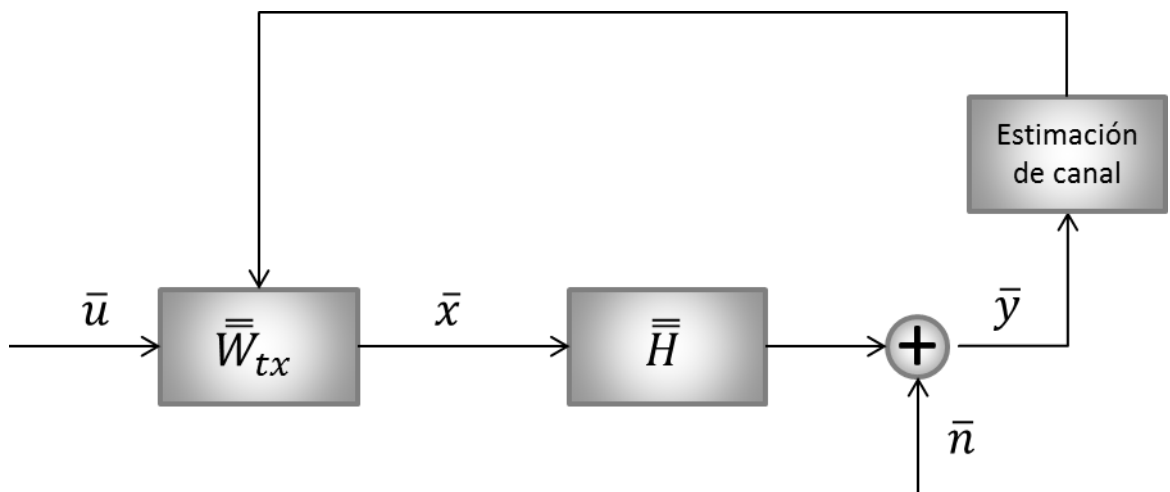


Figura 13. Esquema del precodificador en un sistema MIMO en el canal ascendente.

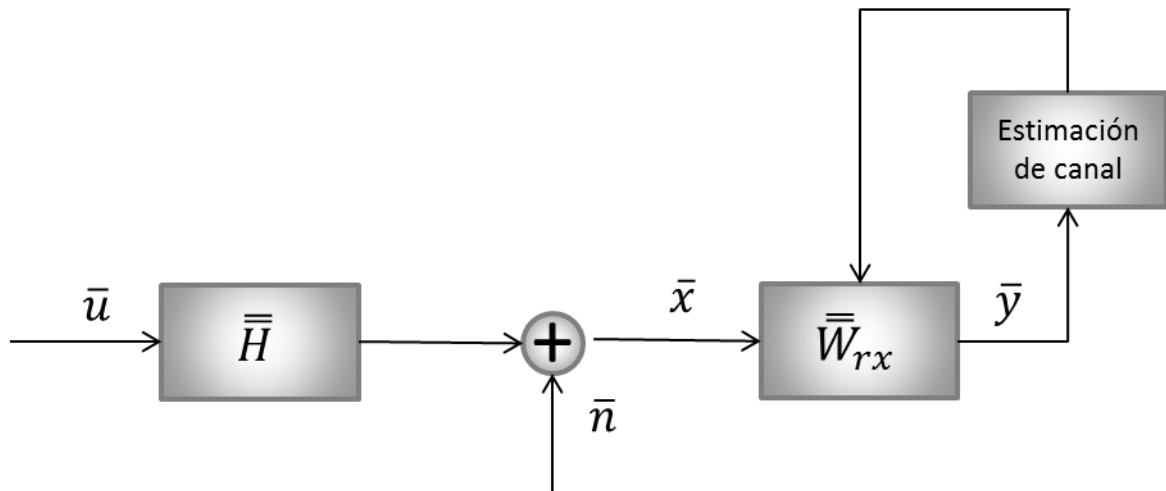


Figura 14. Esquema del filtro receptor en un sistema MIMO en el canal descendente.

Cabe destacar algunos aspectos a tener en cuenta sobre el filtro receptor en el enlace descendente.

En el apartado 1, en la fórmula de la capacidad, podemos observar que se invierte el orden de multiplicación de las matrices ($\bar{H}^* \bar{W}$ pasa a ser $\bar{W}^* \bar{H}$) ya que estamos trabajando en el sentido contrario al ir la señal del receptor al transmisor, y que además, añadimos un operando final ($(\bar{W}^* \bar{W}^H)^{-1}$) como inverso de la matriz de autocorrelación del ruido, ya que en el enlace descendente, el ruido también es filtrado, y puede ser hasta amplificado, teniendo mucho más peso que en el enlace ascendente.

Además, que en el código Matlab que calcula la capacidad del sistema con los diferentes filtros receptores en el enlace descendente, también se invierten las dimensiones de la matriz de canal \bar{H} , siendo M el número de antenas en el receptor y N en el transmisor. También hay que tener en cuenta que en la fórmula de la capacidad, utilizamos la matriz de correlación del que emite, que en el caso del enlace descendente es la BTS, por lo que la traza de esta debe ser N en lugar de M, y para lograr la misma potencia transmitida, hemos de dividir por N.

Por último, cabe destacar que en el enlace descendente, la realimentación no influye en la capacidad, ya que no tenemos que enviar la matriz de canal al transmisor para estimar y podemos asumir que tenemos infinitos bits para representar la matriz de canal \bar{H} .

En todos los escenarios, el filtro calculado va a ser el adaptado con una pequeña modificación en función de cada escenario. El uso del filtro adaptado para someter a estudio los precodificadores se debe a que es el filtro típico y más sencillo de implementar.

4.2. Filtro adaptado

Es el primero de los tres filtros que vamos a utilizar en la precodificación. Simplemente basta con trasponer la matriz de canal \bar{H} .

$$\bar{W} = \bar{H}^H$$

Fórmula 23. Filtro adaptado.

4.3. Filtro adaptado con error

El segundo filtro que vamos a utilizar es similar al anterior, solo que añadiéndole un pequeño error σ debido a la estimación de canal.

Por tanto, definimos el filtro:

$$\bar{W} = \bar{H}^H + \sigma * \bar{E}$$

Fórmula 24. Filtro adaptado con error.

Siendo \bar{E} una matriz aleatoria, gaussiana y compleja donde cada uno de los elementos tiene varianza 1, de las mismas dimensiones que \bar{H} y σ un factor de escala mediante el cual determinamos cómo de grande es el error introducido.

En principio, este filtro debe ser bastante peor que el primero, siendo más parecido a éste cuanto menor sea el error, y difiriendo más con él cuanto mayor sea σ .

4.4. Filtro adaptado con limitación de bits

El tercer y último filtro a utilizar se trata del filtro con limitación de bits.

Como ya hemos dicho antes, tras un primer envío de señal, el receptor capta la matriz de canal $\bar{\bar{H}}$, y hace una estimación de ella. Según el número de bits que utilice para codificar dicha matriz y reenviar al emisor, tenemos más o menos precisión.

Por ejemplo, supongamos que nuestra señal varía entre 0 y 1. Si utilizamos 4 bits para mandar la información (2 para la parte real y dos para la parte imaginaria), sólo tendremos 2^2 niveles de escalado y sólo podremos redondear a 0.25, 0.5, 0.75 o a 1.

Además, establecemos un umbral α , ya que estamos trabajando con una gaussiana, y sólo nos interesa estudiar los valores de atenuación con mayor probabilidad de aparición. Para ello, establecemos una probabilidad de que el valor del coeficiente de atenuación suceda con un 90%.

Una vez que el transmisor ha recibido la información, aplica el filtro adaptado a esa matriz de canal estimada y con valores de amplitud trucados según los niveles definidos por el número de bits utilizados. A partir de este momento, es cuando estudiamos el filtro.

$$\bar{\bar{W}} = \bar{\bar{H}}_{lim}^H$$

Fórmula 25. Filtro limitado en bits.

Este filtro ha de funcionar de tal manera que cuanto mayor sea el número de bits, más se parezca la capacidad obtenida a la que nos proporciona el filtro adaptado, por asemejarse más el filtro con limitación de bits al filtro adaptado.

Capítulo 5

Resultados

En este apartado, incluiremos los gráficos de las simulaciones en Matlab, explicaremos los resultados obtenidos.

Es necesario distinguir entre los enlaces ascendente y descendente ya que difieren en ciertos aspectos.

5.1. Enlace ascendente

En la figura 15, hemos calculado la capacidad del canal ascendente con los diferentes filtros que estamos estudiando. Por ello, utilizamos siempre la misma matriz de canal H y vemos cómo varía la capacidad en función del filtro utilizado.

En ella podemos observar que para una SNR de 20 dB, la máxima capacidad que alcanza el canal es de unos 10 bps/Hz, y se obtiene mediante el uso del filtro adaptado (gráfica 1 de la figura 15).

Cuando el filtro que utilizamos es el que contiene un error de canal (en este caso, σ es 0.1) la capacidad desciende hasta no llegar a alcanzar los 9 bps/Hz. (gráfica 2 de la figura 15).

Por último, en las 3 últimas gráficas de la figura 15, observamos el comportamiento del filtro con limitación de bits. Para un nivel de bits tan bajo como 3, la precisión es pequeña, y tampoco llegamos a alcanzar los 9 bps/Hz. Según vamos aumentando el número de bits, la capacidad va aumentando, hasta conseguir la deseada de 10 bps/Hz al utilizar 10 bits en el estimador.

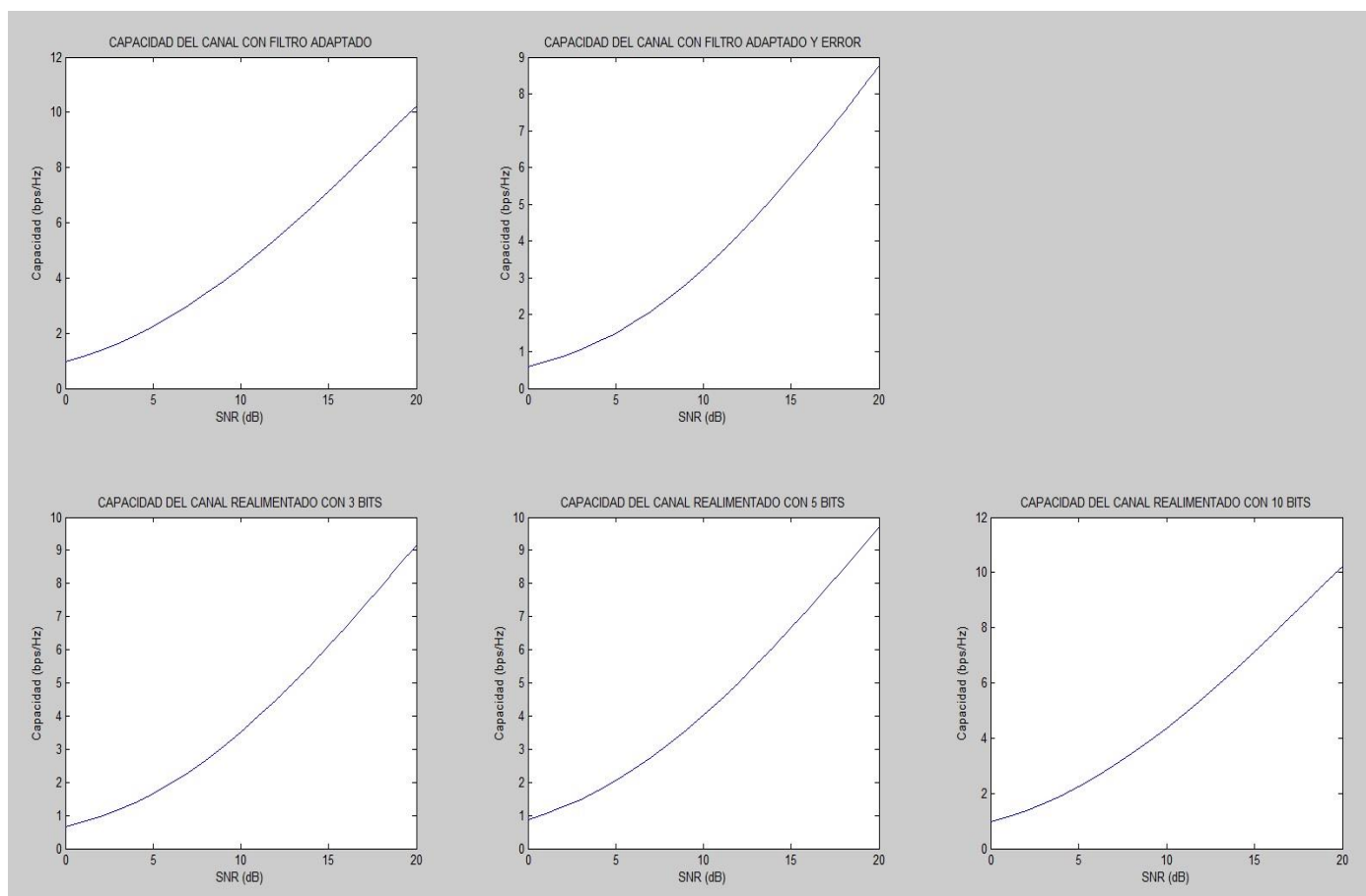


Figura 15. Comparación de los 3 filtros.

A continuación, hemos cogido varias matrices de canal H (unas 100), calculado su capacidad tras pasar por los filtros y representado la media de todas estas medidas para obtener un valor más real del comportamiento de los filtros.

En este caso, lo que hemos variado es el número de antenas que vamos a colocar tanto en recepción como en transmisión, variando estas desde una sola antena en recepción y 20 en transmisión (sistema SIMO), hasta 64 en transmisión y 40 en recepción.

Observamos que el comportamiento es el mismo que hemos explicado un poco más arriba, siendo el filtro adaptado el que mayor capacidad alcanza, seguido del que está limitado a 10 bits (que en este caso, al ser varias medidas y representar la media, no coincide exactamente con el óptimo), y acabando con el filtro con un error de 0.01, y que aún así nos reduce la capacidad casi hasta la mitad.

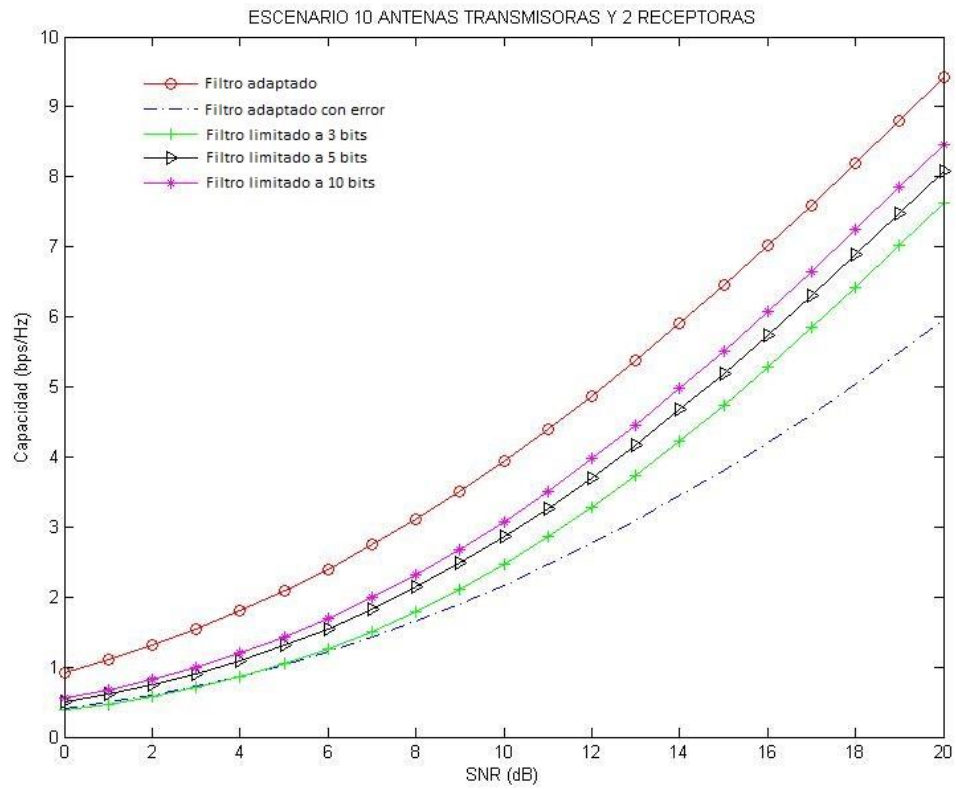


Figura 16. Capacidad del escenario con 10 antenas en transmisión y 2 en recepción.

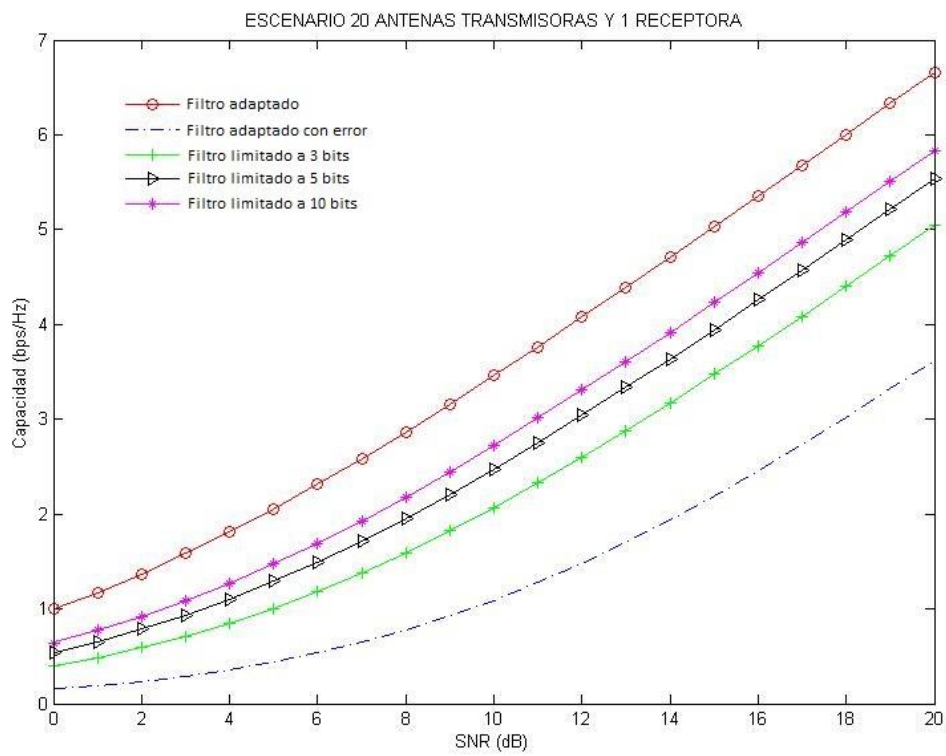


Figura 17. Capacidad del escenario con 20 antenas en transmisión y 1 en recepción.

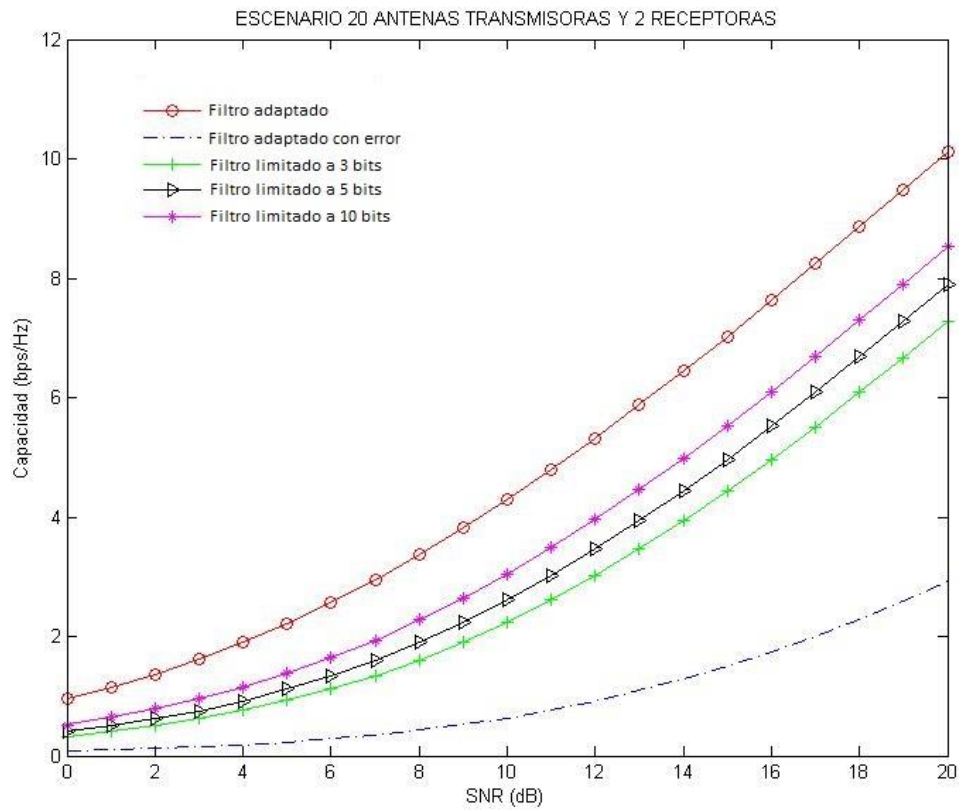


Figura 18. Capacidad del escenario con 20 antenas en transmisión y 2 en recepción.

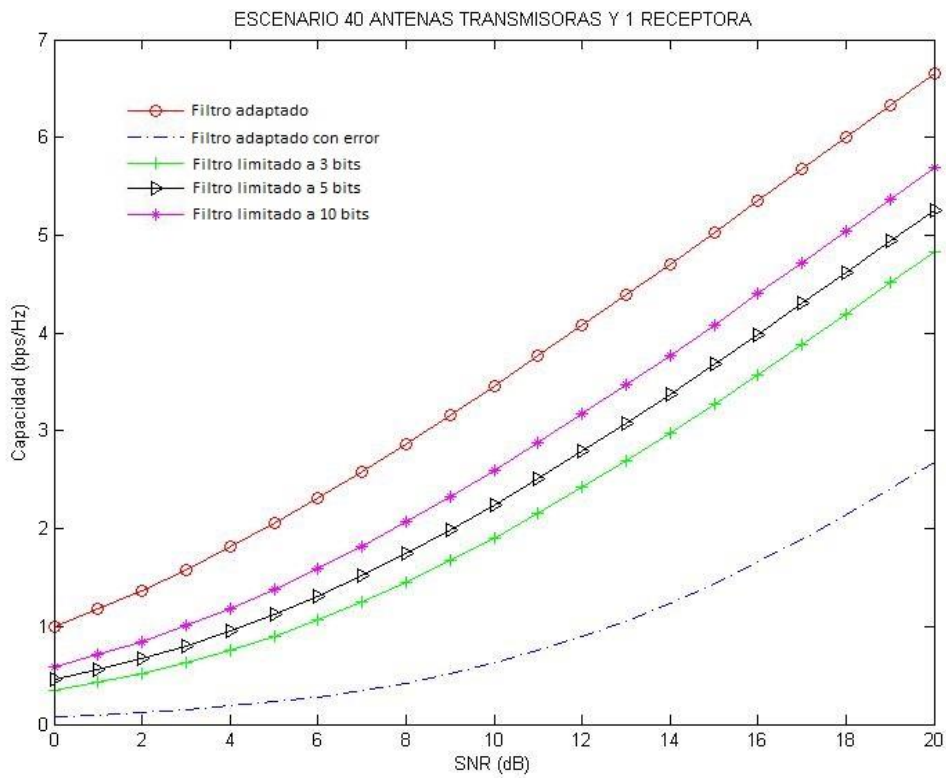


Figura 19. Capacidad del escenario con 40 antenas en transmisión y 1 en recepción.

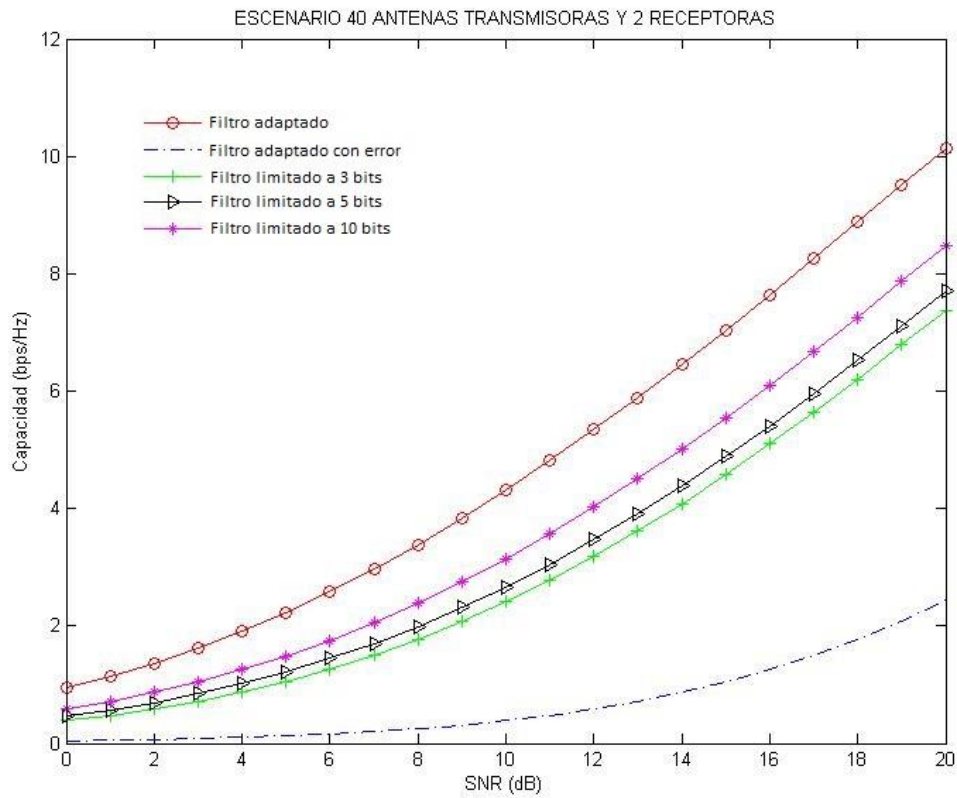


Figura 20. Capacidad del escenario con 40 antenas en transmisión y 2 en recepción.

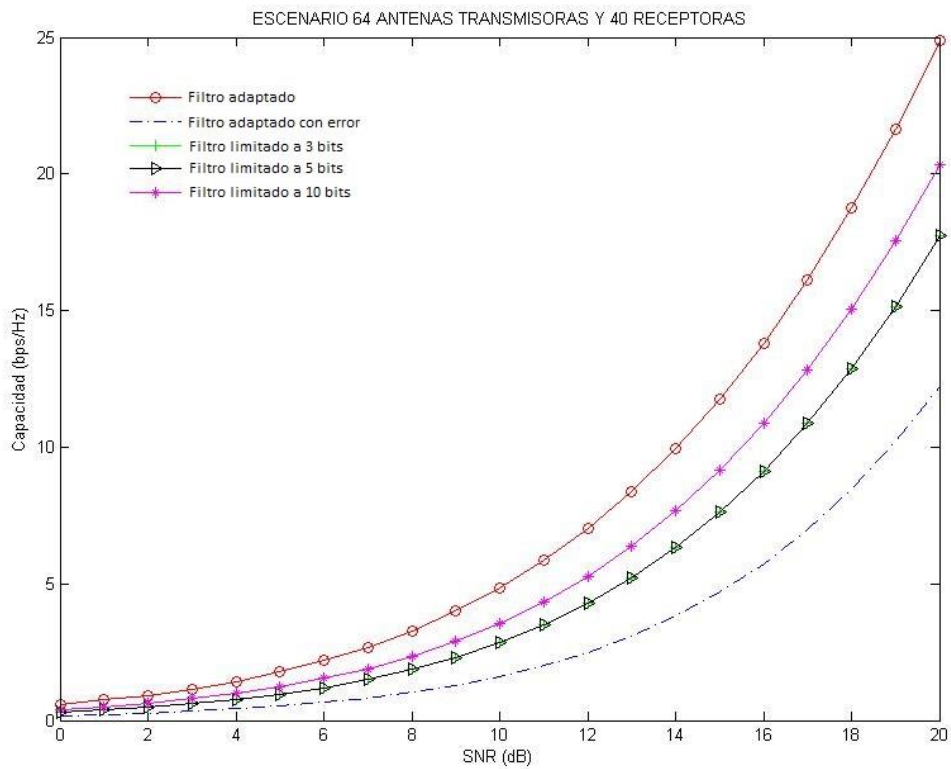


Figura 21. Capacidad del escenario con 64 antenas en transmisión y 40 en recepción.

5.2. Enlace descendente

Tal y como se ha explicado en el capítulo 4, para realizar las simulaciones en el canal descendente, hay que invertir el número de antenas en el transmisor y en el receptor.

En el canal descendente observamos una capacidad mucho más pequeña que en el ascendente. Esto es debido a que en el enlace descendente, el ruido se ve amplificado con el filtro (ya que en este caso si se le aplica) y tiene mucha más fuerza a la hora de calcular la capacidad del canal, lo que hace que disminuya esta al ser inversamente proporcional.

Además, cabe destacar que en este enlace, la capacidad obtenida mediante el uso del filtro adaptado y el uso del filtro limitado a bits coincide siempre. La razón de que esto ocurra es la siguiente. Si el filtro está en el receptor, como en este caso, al hacerse la estimación ahí, no es necesario realimentar la información del canal al transmisor y por tanto no haría falta usar un número finito de bits. Como observábamos en la figura 3, la realimentación en el canal descendente no afecta para nada en la comunicación, ya que la matriz de canal H que enviamos no sufre cambios. Por tanto, lo lógico es que la capacidad sea la misma antes y después de realimentar el sistema.

El método para representar los resultados es el mismo que el utilizado en el canal ascendente.

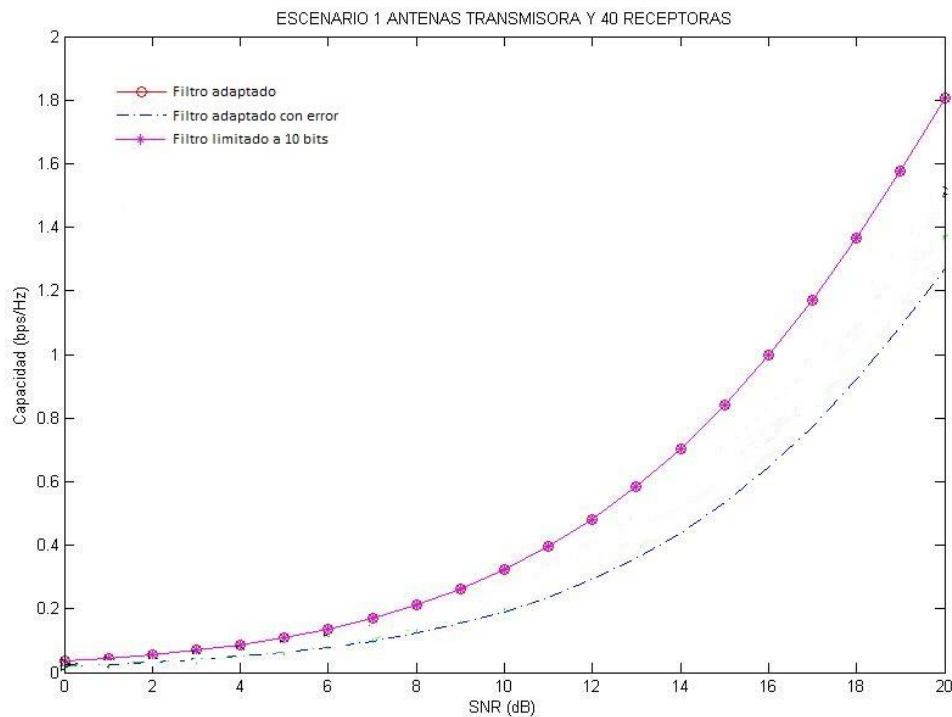


Figura 22. Capacidad del escenario con 1 antena en transmisión y 40 en recepción.

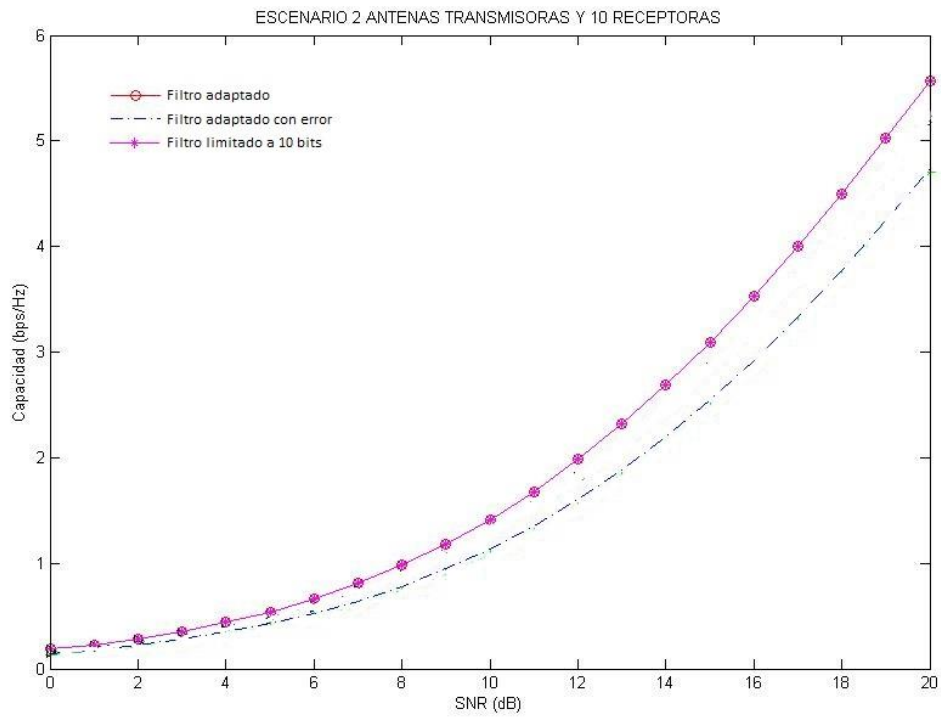


Figura 23. Capacidad del escenario con 2 antenas en transmisión y 10 en recepción.

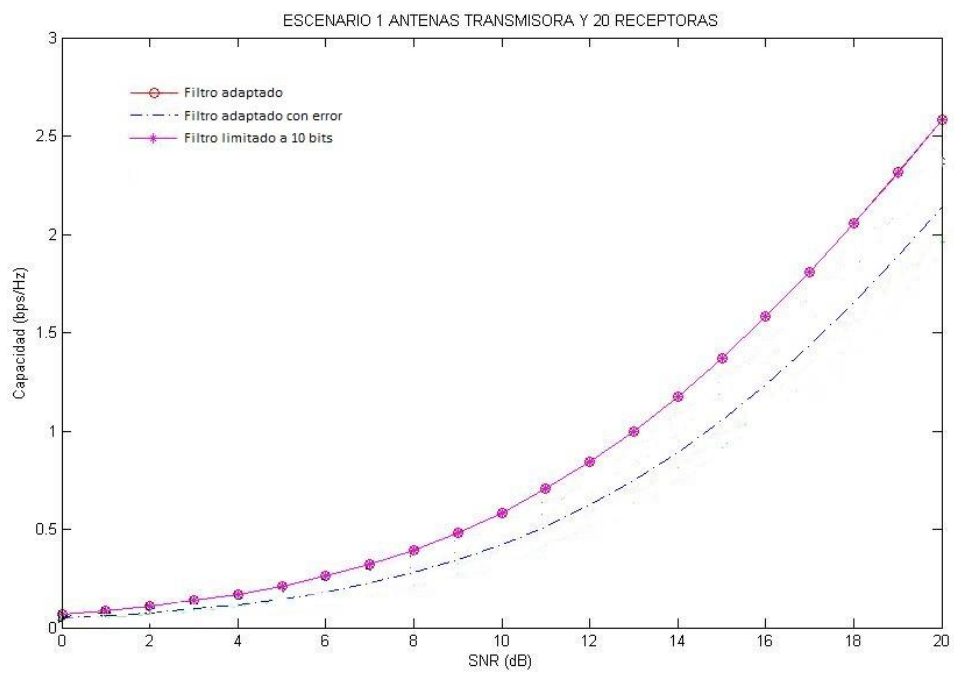


Figura 24. Capacidad del escenario con 1 antena en transmisión y 20 en recepción.

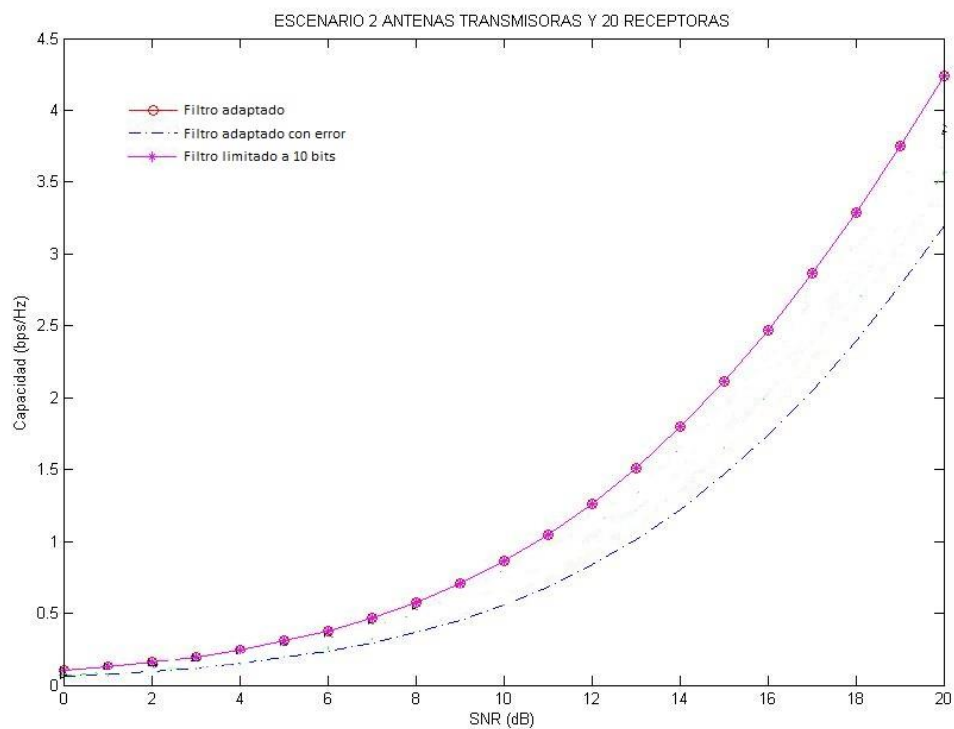


Figura 25. Capacidad del escenario con 2 antenas en transmisión y 20 en recepción.

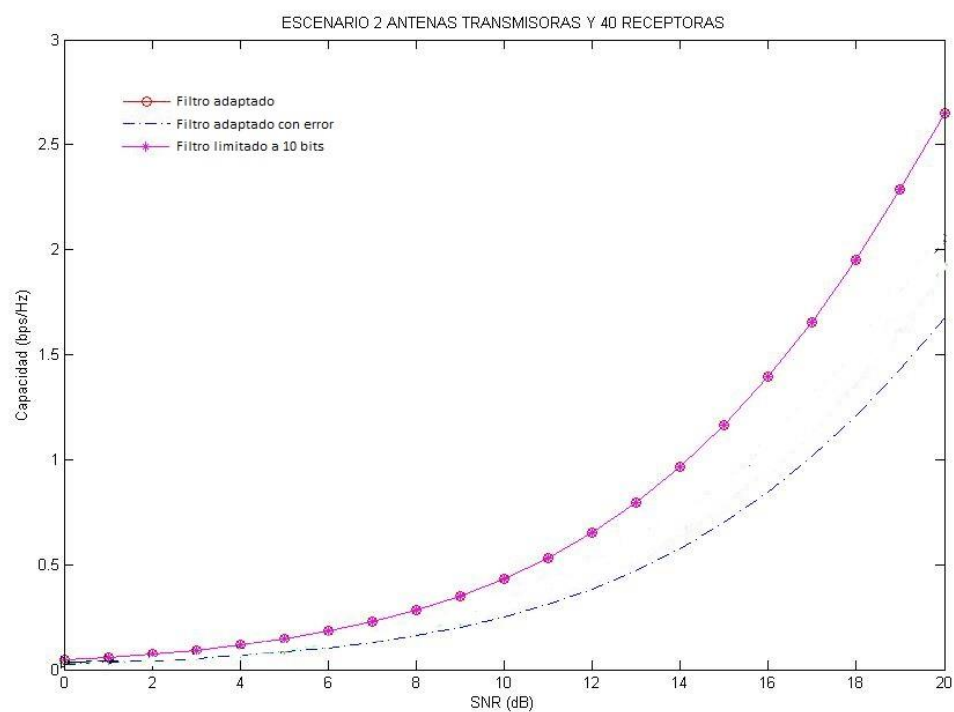


Figura 26. Capacidad del escenario con 2 antenas en transmisión y 40 en recepción.

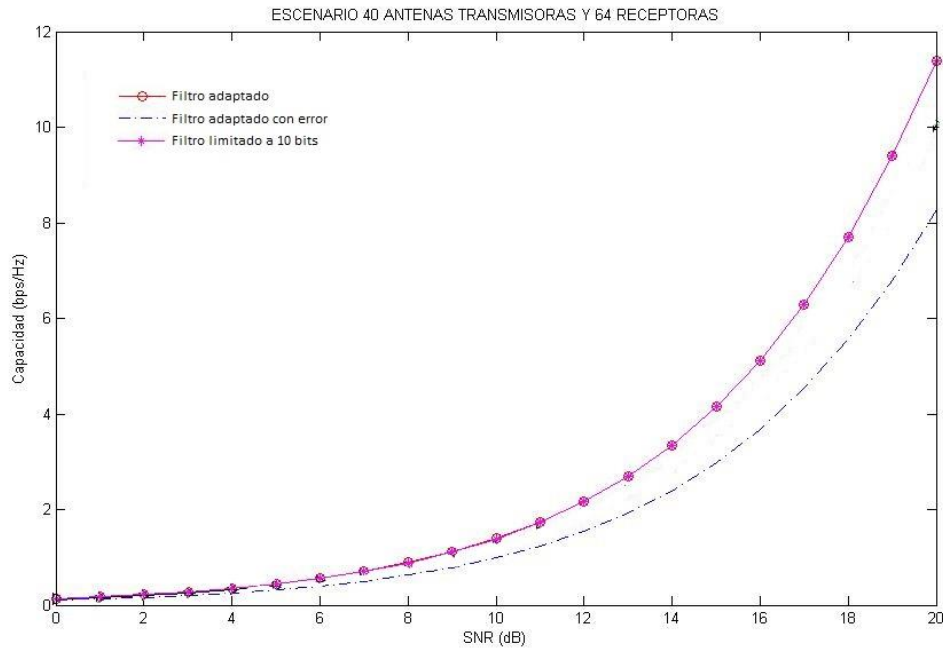


Figura 27. Capacidad del escenario con 40 antenas en transmisión y 64 en recepción.

5.3. Otros estudios realizados

Por último, decidimos estudiar algo más a fondo el filtro estimado con error.

Queremos ver cómo influye el error σ en el comportamiento de este filtro.

Tras las simulaciones, en las cuales espaciábamos más o menos el rango del vector de σ , comprobamos que la capacidad decrece como una exponencial inversa para valores muy bajos de σ (entre 0 y 0.001). Una vez que el error supera este límite, observamos que apenas hay decrecimiento en la capacidad.

Este fenómeno ocurre porque una vez que el error es significativo, el ruido enmascara totalmente a la matriz de canal $\bar{\bar{H}}$, con lo que apenas obtenemos información válida.

A continuación, se muestran estos resultados en las gráficas.

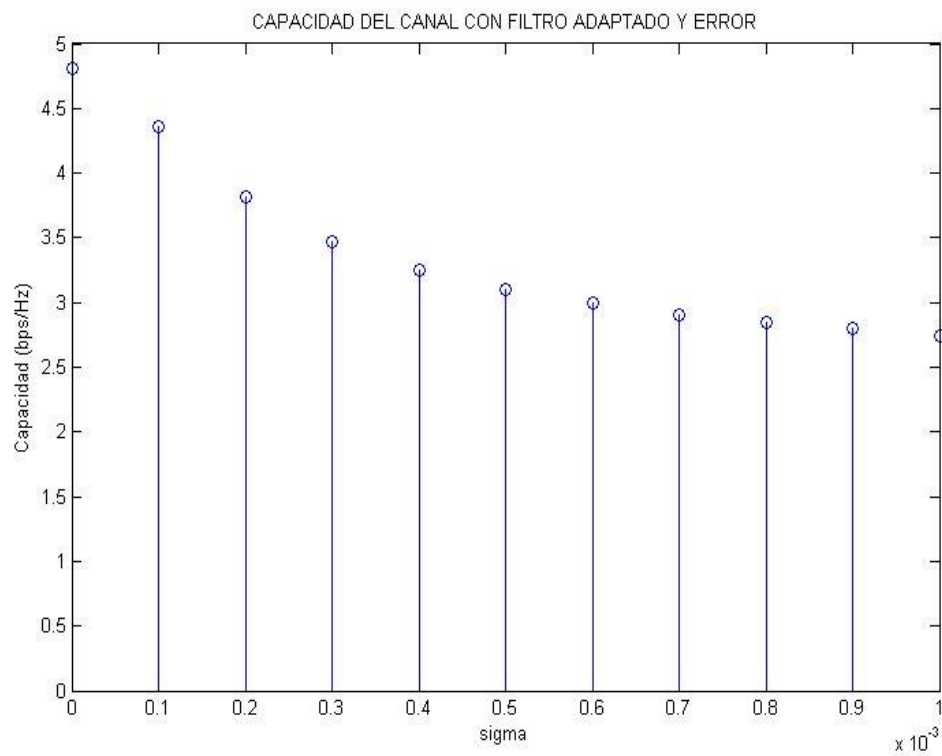


Figura 288. Capacidad del filtro adaptado con error para valores de sigma entre 0 y 0.001.



Figura 19. Capacidad del filtro adaptado con error para valores de sigma entre 0 y 0.1.

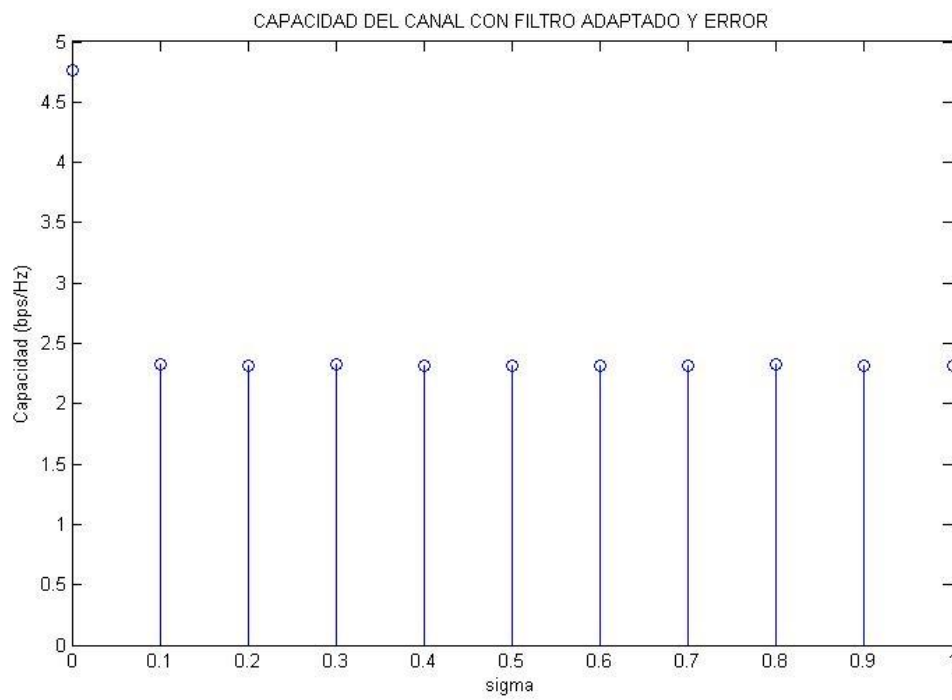


Figura 29. Capacidad del filtro adaptado con error para valores de sigma entre 0 y 1.

Capítulo 6

Líneas futuras

En este trabajo hemos explicado, simulado y analizado 3 posibles escenarios que pueden tener lugar en un sistema MIMO masivo.

Proponemos como posibles continuaciones un análisis más exhausto de las propiedades de estos mismos escenarios (BER, interferencias, ...) con la finalidad de saber con mayor certeza si son escenarios óptimos para llevar a cabo en la realidad o incluso para mejorarlos.

Otra posible salida podría ser mezclar varios de estos escenarios (por ejemplo, un filtro al que primero introduzcamos un error y luego limitemos en bits para ver cómo se comportaría el sistema, si logramos mejorar la capacidad).

$$\bar{\bar{\mathbf{W}}} = (\bar{\bar{\mathbf{H}}}^H + \sigma^2 \bar{\bar{\mathbf{E}}})_{\text{lim}}$$

Fórmula 26. Filtro adaptado con error y posteriormente limitado en bits.

Por último, proponemos estudiar diferentes estrategias de transmisión, tales como el CDI (channel distribution information) y el CSI (channel state information), calculando la, la SVD (singular value decomposition) de la matriz de canal $\bar{\bar{\mathbf{H}}}$ [9].

A continuación explicamos con más detalle esta propuesta.

Si bien el transmisor y el receptor conocen la matriz de canal $\bar{\bar{H}}$, hablamos de un escenario de tipo CSI (channel state information).

La matriz de autocorrelación de la señal de información, $\bar{\bar{R}}_u$, de la fórmula de la capacidad (fórmula 1) es:

$$\bar{\bar{R}}_u = \bar{\bar{V}}^H * \Sigma_p * \bar{\bar{V}}$$

Fórmula 27. Matriz de autocorrelación de la señal de información en CSI o CDI.

Y la matriz $\bar{\bar{H}}$:

$$\bar{\bar{H}} = \bar{\bar{U}}^H * \Sigma * \bar{\bar{V}}$$

Fórmula 28. Matriz H en CSI o CDI.

Siendo en este caso $\bar{\bar{V}}$ la matriz de autovectores por la derecha de la SVD de la matriz $\bar{\bar{H}}$ y Σ_p la matriz diagonal determinista cuya diagonal viene determinada por la potencia del transmisor y del receptor

Por otro lado, si el emisor o el receptor conocen la matriz de autocorrelación de la matriz $\bar{\bar{H}}$, $\bar{\bar{R}}_H$, hablamos de un escenario de tipo CDI (channel distribution information).

Pero en este caso, $\bar{\bar{V}}$ es la SVD de la matriz $\bar{\bar{R}}_H$.

Utilizar la descomposición de matrices en valores singulares es la forma óptima de obtener la mayor capacidad en el canal, debido a la diagonalización.

Capítulo 7

Plan del proyecto

A continuación, detallamos la planificación del proyecto, las fases en que ha sido realizado y el presupuesto que ha sido requerido para su realización.

7.1. Planificación

El proyecto ha tenido una duración total de 10 meses, comprendidos entre Septiembre de 2014 y Junio de 2015. Hemos decidido dividir el proyecto en varias fases que explicamos a continuación.

- Fase 1: Reunión con la tutora, planteamiento de la propuesta y estudio de ésta para comprobar que es adecuada. (Septiembre-Octubre)
- Fase 2: Una vez tomada la decisión de continuar con la propuesta, la tutora plantea los 3 escenarios que van a ser estudiados en el canal ascendente y proporciona a la alumna bibliografía para llevar a cabo el estudio. (Octubre)
- Fase 3: Creación de código Matlab y simulación de los resultados del estudio de capacidad de los 3 escenarios propuestos. (Octubre-Enero)
- Fase 4: Se decide entonces llevar a cabo ese mismo estudio pero en el canal descendente. (Enero)
- Fase 5: Creación de código Matlab y simulación de los resultados del estudio de capacidad de los 3 escenarios propuestos para el canal descendente. (Enero-Marzo)
- Fase 6: Propuesta de nuevos estudios y simulaciones (fijar SNR, variación del error según número de antenas, ...) en los escenarios ya acordados. (Marzo)
- Fase 7: Estudio de estas nuevas propuestas (Marzo-Abril)

- Fase 8: Redacción de la memoria del proyecto y ensayos de la presentación. (Abril-Junio)

7.2. Presupuesto

Se detalla a continuación el presupuesto de este trabajo, dividido entre el trabajo como ingenieros del profesor y del alumno (mano de obra) y el coste de los recursos utilizados (material).

Mano de obra:

El alumno ha trabajado alrededor de 10 horas semanales a lo largo de 40 semanas, lo que hace un total de 400 horas. El profesor ha trabajado dos horas semanales durante 40 semanas, lo que hace un total de 80 horas.

Suponemos un sueldo básico para el alumno de 1400 euros al mes, lo que nos da como resultado 8 € por hora. Suponemos un sueldo básico para el tutor de 3000 euros al mes, lo que da como resultado 17€ por hora.

- Alumno: 400 horas x 8€/hora → 3200€.
- Tutor: 80 horas x 17€/hora → 1360€.

- **Total mano de obra → 4560€.**

Material:

- Portátil: 400 €.
- Licencia de MATLAB: 2000 €.

- **Total material → 2400 €.**

Presupuesto final (mano de obra + material) → 6960€.

Capítulo 8

Conclusions

The results that we have obtained are the ones that we expect before starting this Project:

- Using massive MIMO in real life is a good idea. We have already prove that this system increases its capacity.
- We can order the three filters that we study in order to his efficiency:

1º *Matched filter*: The one that provide us the biggest capacity.

2º *Limited bit filter*: This is the most similar to the first one. We conclude that delimiting this filter with 10 bits is enough to obtain almost the same capacity that the matched one, and while we reduce the number of bits, the capacity separates more from the maximum possible.

3º *Adapted filter with error*: However small the error is, this filter never get the capacity of the adapted filter.

- Because the downlink does not have feedback, both adapted filter and limited filter get the same capacity.
- When the error of the second filter overpass the 10^{-3} level, the capacity does not change anymore, because the filter saturates.

Capítulo 9

Conclusiones

Todos los resultados que hemos obtenido han sido los que esperábamos al comienzo del proyecto:

- El uso de MIMO masivo en la vida real es factible y se ha demostrado que aumenta la capacidad de este.
- El orden de eficacia de los tres filtros utilizados es el siguiente:
 - 1º *Filtro adaptado*: Es el que mayor capacidad de canal nos proporciona.
 - 2º *Filtro con limitación de bits*: Es el que más se asemeja al primero. Hemos determinado que con una limitación de 10 bits obtenemos prácticamente la misma capacidad que con el filtro adaptado, y que según disminuimos el número de bits, la capacidad se va alejando de la máxima posible.
 - 3º *Filtro adaptado con error*: Por muy pequeño que sea el error, este filtro es el que peores prestaciones no ha ofrecido.
- La no-realimentación en el canal descendente hace que tanto el filtro adaptado como el filtro delimitado en bits tengan exactamente el mismo comportamiento, ya que el emisor no aprende.
- El error introducido en el filtro adaptado acaba saturando, haciendo que si este está comprendido entre 0 y 10^{-3} se note la bajada de prestaciones, pero que llegado al nivel de 10^{-3} , la capacidad deje de disminuir y permanezca constante.

Anexos

Código Matlab

Canal_jj.m

```
% Generacion de la matriz de canal
% 06/04/2005 Utilizando matriz de scattering
% Genera la matriz de canal normalizada en potencia con un determinado
ASD y ASA
% Está validado que cuando ambos tienden a 0 llego a Free Space,
salvo por una fase constate que viene dada por la variabla aleatoria
gausiana por la que se multiplica

function [H]=canal(lambda,D,dt,dr,teta,ASD,ASA,Hg,M,N)

    step=0.4;
    [Tx, Rx] = posicion_tx_rx(M, N, teta, D, dt, dr);

    [M index]=size(Tx);
    [N index]=size(Rx);

    teta_tx=0;
    teta_rx=teta;

    phi=sort([teta_rx-teta_tx-pi:-step*pi/180:-pi teta_rx-
teta_tx:step*pi/180:pi teta_rx-teta_tx-
pi+step*pi/180:step*pi/180:teta_rx-teta_tx-step*pi/180]);
    L=length(phi);
    k=2*pi*[real(exp(sqrt(-1)*phi));imag(exp(sqrt(-1)*phi))]/lambda;

    P_L=circshift(ASD/sqrt(2)*exp(-
sqrt(2)*abs(phi')/ASD),find(phi==teta_rx-teta_tx)-floor(L/2)-1)';
    P_L=P_L/sum(P_L);

    P_u=zeros(1,L);
    P_u( find(((circshift(phi',find(phi==teta_rx-teta_tx)-floor(L/2)-
1)')>=-ASA/2)&((circshift(phi',find(phi==teta_rx-teta_tx)-floor(L/2)-
1)')<=ASA/2)))=1/ASA;
    P_u=P_u/sum(P_u);

    Bf_tx = exp(-1i * mod((Tx * k').', 2*pi));
    Bf_rx = exp(1i * mod((Rx * k'), 2*pi));

    H=Bf_rx*diag(sqrt(P_u))*Hg*diag(sqrt(P_L))*Bf_tx;

end
```

posicion_tx_rx.m

```
% Funcion que devuelve los arrays de posicion de las antenas en
transmisi3n y recepci3n

function [Tx_vector, Rx_vector]=posicion_tx_rx(M, N, teta, D, dt, dr)

% ----- Parametros -----%
% M:          numero de antenas en tx
% N:          numero de antenas en rx
% teta:       Angulo entre eje perpendicular de tx y eje perpendicular
de rx.
%             (Como de alineados estan los arrays de antenas)
% D:          distancia entre arrays de transmisi3n y recepci3n
% dt:         distancia entre antenas en transmisi3n
% dr:         distancia entre antenas en recepci3n
% -----%

% Debemos diferenciar entre M y N impar o par, respectivamente
% Si par: No antena en el origen
% Si no par: Antena en el origen

% ----- TRANSMISION -----
% Suponemos antenas en transmisi3n en el origen

origen = zeros(M, 1);

if rem(M, 2) == 0
    % Numero par de antenas en transmisi3n
    antenas_negativas = zeros(M/2, 1);
    antenas_positivas = zeros(M/2, 1);
    antenas_negativas(1) = -1;
    antenas_positivas(M/2) = 1;

    for i = 2:M/2
        antenas_negativas(i) = -2*i + 1;
        antenas_positivas(M/2-i+1) = 2*i - 1;
    end

    Tx_vector = [antenas_positivas; antenas_negativas];
    Tx_vector = [origen Tx_vector];
else
    % Numero impar de antenas en transmisi3n
    antenas_negativas = zeros((M-1)/2, 1);
    antenas_positivas = zeros((M-1)/2, 1);
    for i = 1:(M-1)/2
        antenas_negativas(i) = -2*i;
        antenas_positivas((M-1)/2-i+1) = 2*i;
    end

    Tx_vector = [antenas_positivas; 0; antenas_negativas];
    Tx_vector = [origen Tx_vector];
end

Tx_vector = Tx_vector*dt/2;

% ----- RECEPCION -----
% Posici3n central del array recepci3n
```

```

R=D*[cos(teta) sin(teta)];
R = ones(N, 1)*R;

if rem(N, 2) == 0
    % Numero par de antenas en recepcion
    antenas_negativas = zeros(N/2, 1);
    antenas_positivas = zeros(N/2, 1);
    antenas_negativas(1) = -1;
    antenas_positivas(N/2) = 1;

    for i = 2:N/2
        antenas_negativas(i) = -2*i + 1;
        antenas_positivas(N/2-i+1) = 2*i - 1;
    end

    Rx_col = [antenas_positivas; antenas_negativas];
    col1 = -Rx_col*sin(teta);
    col2 = Rx_col*cos(teta);
    Rx_vector = [col1 col2];
    Rx_vector = Rx_vector*dr/2;
    %Anyadimos la distancia a la que se encuentra este array
    Rx_vector = R + Rx_vector;

else
    % Numero impar de antenas en recepcion
    antenas_negativas = zeros((N-1)/2, 1);
    antenas_positivas = zeros((N-1)/2, 1);
    for i = 1:(N-1)/2
        antenas_negativas(i) = -2*i;
        antenas_positivas((N-1)/2-i+1) = 2*i;
    end

    Rx_col = [antenas_positivas; 0; antenas_negativas];
    col1 = -Rx_col*sin(teta);
    col2 = Rx_col*cos(teta);
    Rx_vector = [col1 col2];
    Rx_vector = Rx_vector*dr/2;
    %Anyadimos la distancia a la que se encuentra este array
    Rx_vector = R + Rx_vector;

end

end
end

```

CanalH.m

% Función que calcula la matriz de canal H normalizada a partir de unos parámetros que se explican a continuación, elegidos apropiadamente para conseguir que dicha matriz sea ortogonal.

```
function [Hnorm] = canalH(M,N)
```

```
% -----%  
% ----- PARÁMETROS -----%  
% -----%  
% lambda: Longitud de onda.  
% D: Distancia entre arrays de transmisión y recepción.  
% dt: Distancia entre antenas en transmisión.  
% dr: Distancia entre antenas en recepción.  
% teta: Inclinación del array de recepción con respecto al eje  
% perpendicular del array de transmisión.  
% ASD: Dispersión angular en el transmisor.  
% ASA: Dispersión angular en el receptor.  
% Hg: Matriz de canal gaussiana.  
% M: Número de antenas en transmisión.  
% N: Número de antenas en recepción.  
% -----%  
% -----%
```

```
lambda=0.15;  
D=1000;  
dt=0.5*lambda;  
dr=0.5*lambda;  
teta=0;  
ASD=30*pi/180;  
ASA=360*pi/180;  
L=901;  
Hg=(randn(L,L)+sqrt(-1)*randn(L,L))/sqrt(2);  
M = 40;  
N = 64;  
  
H=canal_jj(lambda,D,dt,dr,teta,ASD,ASA,Hg,M,N);  
  
Hnorm=H/norm(H);
```

```
end
```

MatchedFilter.m

```
% Función que calcula el primero de los tres filtros que vamos a
comparar, así como su capacidad en función de la SNR.
% Se trata del filtro adaptado.
```

```
function [SNR,Rm] = MatchedFilter()

    close all
    clear all
    clc

    for i=1:10

        H=canalH();

        M=size(H,2);
        N=size(H,1);
        w=H';
        I=eye(N);
        SNR=[0:1:20];
        SNRnat=10.^(SNR/10);

        for j=1:size(SNR,2)
            R(j,i)=log2(det(I+((SNRnat(j)/M)*H*(w*w')*H')));
        end

    end

    Rm=mean(R,2);
    figure, plot(SNR,Rm)

end
```

MatchedFilterError.m

% Función que calcula el segundo de los tres filtros que vamos a comparar, así como su capacidad en función de la SNR.
% Se trata del filtro adaptado añadiéndole un pequeño error.

```
function [Rm] = MatchedFilterError()

    close all
    clear all
    clc

    for i=1:10

        H=canalH();
        M=size(H,2);
        N=size(H,1);

        I=eye(N);

        SNR=[0:1:20];
        SNRnat=10.^(SNR/10);

        sigma=0.01;
        He=H+(sigma*(randn(N,M)+i*randn(N,M)));
        w=He';
        w=w./sqrt(trace(w*w')/M);

        for j=1:size(SNR,2)
            R(j,i)=log2(det(I+((SNRnat(j)/M)*H*(w*w')*H')));
        end

    end

    Rm=mean(R,2);
    figure, plot(SNR,Rm)

end
```


SampleFilter3bit.m

% Función que calcula el tercer filtro de los que vamos a comparar,
% así como su capacidad en función de la SNR.
% Se trata del filtro limitado en bits. En este caso usamos 3 bits
para escalar, con lo que conseguimos 4 niveles de escalado.

```
function [Rm] = SampleFilter3bit()

close all
clear all
clc

for i=1:10

    H=canalH();
    M=size(H,2);
    N=size(H,1);
    I=eye(N);
    SNR=[0:1:20];
    SNRnat=10.^(SNR/10);
    alpha=qfuncinv(0.01)*(1/sqrt(2));
    Nbits=3;

    for j=1:M

        for k=1:N

            valores=[-alpha:(2*alpha)/((2^Nbits)/2)-1):alpha];

            for l=1:size(valores,2)

                distR(l)=real(H(k,j))-valores(l);
                distI(l)=imag(H(k,j))-valores(l);

            end

            [valorR,indR]=min(distR);
            [valorI,indI]=min(distI);
            Hest(k,j)=valores(indR)+i*valores(indI);

        end

    end

    w=Hest';
    w=w./sqrt(trace(w*w')/M);

    for j=1:size(SNR,2)
        R(j,i)=log2(det(I+((SNRnat(j)/M)*H*(w*w')*H')));
    end

end

Rm=mean(R,2);
figure, plot(SNR,Rm)

end
```

SampleFilter5bit.m

```
% Función que calcula el tercer filtro de los que vamos a comparar,  
% así como su capacidad en función de la SNR.  
% Se trata del filtro limitado en bits. En este caso usamos 5 bits  
% para escalar, con lo que conseguimos 16 niveles de escalado.
```

```
function [Rm] = SampleFilter5bit()  
  
close all  
clear all  
clc  
  
for i=1:10  
  
    H=canalH();  
    M=size(H,2);  
    N=size(H,1);  
    I=eye(N);  
    SNR=[0:1:20];  
    SNRnat=10.^(SNR/10);  
    alpha=qfuncinv(0.01)*(1/sqrt(2));  
    Nbits=5;  
  
    for j=1:M  
  
        for k=1:N  
  
            valores=[-alpha:(2*alpha)/((2^Nbits)/2)-1):alpha];  
  
            for l=1:size(valores,2)  
  
                distR(l)=real(H(k,j))-valores(l);  
                distI(l)=imag(H(k,j))-valores(l);  
  
            end  
  
            [valorR,indR]=min(distR);  
            [valorI,indI]=min(distI);  
  
            Hest(k,j)=valores(indR)+i*valores(indI);  
  
        end  
  
    end  
  
    w=Hest';  
    w=w./sqrt(trace(w*w')/M);  
  
    for j=1:size(SNR,2)  
        R(j,i)=log2(det(I+((SNRnat(j)/M)*H*(w*w')*H')));  
    end  
  
end  
  
Rm=mean(R,2);  
figure, plot(SNR,Rm)  
  
end
```

SampleFilter10bit.m

% Función que calcula el tercer filtro de los que vamos a comparar,
% así como su capacidad en función de la SNR.
% Se trata del filtro limitado en bits. En este caso usamos 10 bits
para escalar, con lo que conseguimos 512 niveles de escalado.

```
function [Rm] = SampleFilter10bit()

close all
clear all
clc

for i=1:10

    H=canalH();
    M=size(H,2);
    N=size(H,1);
    I=eye(N);
    SNR=[0:1:20];
    SNRnat=10.^(SNR/10);
    alpha=qfuncinv(0.01)*(1/sqrt(2));
    Nbits=10;

    for j=1:M

        for k=1:N

            valores=[-alpha:(2*alpha)/((2^Nbits)/2)-1):alpha];

            for l=1:size(valores,2)

                distR(l)=real(H(k,j))-valores(l);
                distI(l)=imag(H(k,j))-valores(l);

            end

            [valorR,indR]=min(distR);
            [valorI,indI]=min(distI);

            Hest(k,j)=valores(indR)+i*valores(indI);

        end

    end

    w=Hest';
    w=w./sqrt(trace(w*w')/M);

    for j=1:size(SNR,2)
        R(j,i)=log2(det(I+((SNRnat(j)/M)*H*(w*w')*H')));
    end

end

Rm=mean(R,2);
figure, plot(SNR,Rm)

end
```

Compare.m

% Función que para una misma matriz de canal H calcula su capacidad tras pasar por todos los filtros para poder comparar.

```
function [] = Compare()

close all
clear all
clc

H=canalH();
M=size(H,2);
N=size(H,1);
I=eye(N);
SNR=[0:1:20];
SNRnat=10.^(SNR/10);
alpha=qfuncinv(0.01)*(1/sqrt(2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO ADAPTADO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w1=H';
w1=w1./sqrt(trace(w1*w1')/M);

for j=1:size(SNR,2)
    R1(j)=log2(det(I+((SNRnat(j)/M)*H*(w1*w1')*H')));
end

subplot(2,3,1)
plot(SNR,real(R1));
title('CAPACIDAD DEL CANAL CON FILTRO ADAPTADO')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO ADAPTADO CON ERROR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sigma=0.1;
He=H+(sigma*(randn(N,M)+i*randn(N,M)));
w2=He';
w2=w2./sqrt(trace(w2*w2')/M);

for j=1:size(SNR,2)
    R2(j)=log2(det(I+((SNRnat(j)/M)*H*(w2*w2')*H')));
end

subplot(2,3,2)
plot(SNR,real(R2));
title('CAPACIDAD DEL CANAL CON FILTRO ADAPTADO Y ERROR')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 3 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nbits1=3;

for j=1:M

    for k=1:N
```

```

valores1=[-alpha:(2*alpha)/((2^Nbits1)/2)-1):alpha];

for l=1:size(valores1,2)

    distR1(l)=real(H(k,j))-valores1(l);
    distI1(l)=imag(H(k,j))-valores1(l);

end

[valorR1,indR1]=min(abs(distR1));
[valorI1,indI1]=min(abs(distI1));

Hest(k,j)=valores1(indR1)+i*valores1(indI1);

end

end

w3=Hest';
w3=w3./sqrt(trace(w3*w3')/M);

for j=1:size(SNR,2)
    R3(j)=log2(det(I+((SNRnat(j)/M)*H*(w3*w3')*H')));
end

subplot(2,3,4)
plot(SNR,real(R3));
title('CAPACIDAD DEL CANAL REALIMENTADO CON 3 BITS')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')

%%%%%%%%%%%%%% FILTRO LIMITADO A 5 BITS %%%%%%%%%%%%%%%

Nbits2=5;

for j=1:M

    for k=1:N

        valores2=[-alpha:(2*alpha)/((2^Nbits2)/2)-1):alpha];

        for l=1:size(valores2,2)

            distR2(l)=real(H(k,j))-valores2(l);
            distI2(l)=imag(H(k,j))-valores2(l);

        end

        [valorR2,indR2]=min(abs(distR2));
        [valorI2,indI2]=min(abs(distI2));

        Hest2(k,j)=valores2(indR2)+i*valores2(indI2);

    end

end

end

w4=Hest2';
w4=w4./sqrt(trace(w4*w4')/M);

for j=1:size(SNR,2)

```

```

        R4(j)=log2(det(I+((SNRnat(j)/M)*H*(w4*w4')*H')));
    end

    subplot(2,3,5)
    plot(SNR,real(R4));
    title('CAPACIDAD DEL CANAL REALIMENTADO CON 5 BITS')
    xlabel('SNR (dB)')
    ylabel('Capacidad (bps/Hz)')

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 10 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    Nbits3=10;

    for j=1:M

        for k=1:N

            valores3=[-alpha:(2*alpha)/((2^Nbits3)/2)-1):alpha];

            for l=1:size(valores3,2)

                distR3(l)=real(H(k,j))-valores3(l);
                distI3(l)=imag(H(k,j))-valores3(l);

            end

            [valorR3,indR3]=min(abs(distR3));
            [valorI3,indI3]=min(abs(distI3));

            Hest3(k,j)=valores3(indR3)+i*valores3(indI3);

        end

    end

    w5=Hest3';
    w5=w5./sqrt(trace(w5*w5')/M);

    for j=1:size(SNR,2)
        R5(j)=log2(det(I+((SNRnat(j)/M)*H*(w5*w5')*H')));
    end

    subplot(2,3,6)
    plot(SNR,real(R5));
    title('CAPACIDAD DEL CANAL REALIMENTADO CON 10 BITS')
    xlabel('SNR (dB)')
    ylabel('Capacidad (bps/Hz)')

end

```

Ascendente.m

```
% Función que genera 100 matrices de canal H, calcula sus capacidades  
% tras pasar por los diferentes filtros y representa la media de estas  
% capacidades.  
% Enlace ascendente.
```

```
function [] = Ascendente()
```

```
close all  
clear all  
clc
```

```
for i=1:100
```

```
    H=canalH();  
    M=size(H,2);  
    N=size(H,1);  
    I=eye(N);  
    SNR=[0:1:20];  
    SNRnat=10.^(SNR/10);  
    alpha=qfuncinv(0.01)*(1/sqrt(2));
```

```
    %%%%%%%%%%%%%%% FILTRO ADAPTADO %%%%%%%%%%%%%%%
```

```
        w1=H';  
        w1=w1./sqrt(trace(w1*w1')/M);  
  
        for j=1:size(SNR,2)  
            R1(j)=log2(det(I+((SNRnat(j)/M)*H*(w1*w1')*H')));  
        end  
  
        Rm1=mean(R1,2);
```

```
    %%%%%%%%%%%%%%% FILTRO ADAPTADO CON ERROR %%%%%%%%%%%%%%%
```

```
        sigma=0.1;  
        He=H+(sigma*(randn(N,M)+i*randn(N,M)));  
        w2=He';  
        w2=w2./sqrt(trace(w2*w2')/M);  
  
        for j=1:size(SNR,2)  
            R2(j)=log2(det(I+((SNRnat(j)/M)*H*(w2*w2')*H')));  
        end  
  
        Rm2=mean(R2,2);
```

```
    %%%%%%%%%%%%%%% FILTRO LIMITADO A 3 BITS %%%%%%%%%%%%%%%
```

```
    Nbits1=3;
```

```
    for j=1:M
```

```
        for k=1:N
```

```
            valores1=[-alpha:(2*alpha)/((2^Nbits1)/2)-  
1):alpha];
```

```
            for l=1:size(valores1,2)
```

```

        distR1(l)=real(H(k,j))-valores1(l);
        distI1(l)=imag(H(k,j))-valores1(l);

    end

    [valorR1,indR1]=min(abs(distR1));
    [valorI1,indI1]=min(abs(distI1));

    Hest(k,j)=valores1(indR1)+i*valores1(indI1);

end

end

w3=Hest';
w3=w3./sqrt(trace(w3*w3')/M);

for j=1:size(SNR,2)
    R3(j)=log2(det(I+((SNRnat(j)/M)*H*(w3*w3')*H')));
end

Rm3=mean(R3,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 5 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nbits2=5;

for j=1:M

    for k=1:N

        valores2=[-alpha:(2*alpha)/((2^Nbits2)/2)-
1):alpha];

        for l=1:size(valores2,2)

            distR2(l)=real(H(k,j))-valores2(l);
            distI2(l)=imag(H(k,j))-valores2(l);

        end

        [valorR2,indR2]=min(abs(distR2));
        [valorI2,indI2]=min(abs(distI2));

        Hest2(k,j)=valores2(indR2)+i*valores2(indI2);

    end

end

w4=Hest2';
w4=w4./sqrt(trace(w4*w4')/M);

for j=1:size(SNR,2)
    R4(j)=log2(det(I+((SNRnat(j)/M)*H*(w4*w4')*H')));
end

Rm4=mean(R4,2);

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 10 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nbits3=10;

for j=1:M

    for k=1:N

        valores3=[-alpha:(2*alpha)/((2^Nbits3)/2)-
1):alpha];

        for l=1:size(valores3,2)

            distR3(l)=real(H(k,j))-valores3(l);
            distI3(l)=imag(H(k,j))-valores3(l);

        end

        [valorR3,indR3]=min(abs(distR3));
        [valorI3,indI3]=min(abs(distI3));

        Hest3(k,j)=valores3(indR3)+i*valores3(indI3);

    end

end

w5=Hest3';
w5=w5./sqrt(trace(w5*w5')/M);

for j=1:size(SNR,2)
    R5(j)=log2(det(I+((SNRnat(j)/M)*H*(w5*w5')*H')));
end

Rm5=mean(R5,2);

end

figure
plot(SNR,R1,'-ro',SNR,R2,'-.b',SNR,R3,'-+g',SNR,R4,'->k',SNR,R5,'-
*m')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')
legend('Matched filter','Matched filter error','3 bit sample
filter','5 bit sample filter','10 bit sample filter')

end

```

Descendente.m

```
% Función que genera 100 matrices de canal H, calcula sus capacidades  
% tras pasar por los diferentes filtros y representa la media de estas  
% capacidades.  
% Enlace descendente.
```

```
function [] = Descendente()
```

```
close all  
clear all  
clc
```

```
for x=1:100
```

```
    H=canalH();  
    M=size(H,2);  
    N=size(H,1);  
    I=eye(M);  
    SNR=[0:1:20];  
    SNRnat=10.^(SNR/10);  
    alpha=qfuncinv(0.01)*(1/sqrt(2));
```

```
    %%%%%%%%%%%%%%% FILTRO ADAPTADO %%%%%%%%%%%%%%%
```

```
        w1=H';
```

```
        for j=1:size(SNR,2)  
            % R1(j)=log2(det(I+(SNRnat(j)/M)*H*(w1*w1')*H'));
```

```
        R1(j)=log2(det(I+(SNRnat(j)/N)*(w1*H)*(H'*w1')*inv(w1*w1')));  
        end
```

```
        Rm1=mean(R1,2);
```

```
    %%%%%%%%%%%%%%% FILTRO ADAPTADO CON ERROR %%%%%%%%%%%%%%%
```

```
        sigma=0.1;  
        He=H+(sigma*(randn(N,M)+i*randn(N,M)));  
        w2=He';
```

```
        for j=1:size(SNR,2)
```

```
            R2(j)=log2(det(I+(SNRnat(j)/N)*((w2*H)*(H'*w2'))*inv(w2*w2')));  
            end
```

```
            Rm2=mean(R2,2);
```

```
    %%%%%%%%%%%%%%% FILTRO LIMITADO A 3 BITS %%%%%%%%%%%%%%%
```

```
        Nbits1=3;
```

```
        for j=1:M
```

```
            for k=1:N
```

```
                valores1=[-alpha:(2*alpha)/((2^Nbits1)/2)-  
1):alpha];
```

```

        for l=1:size(valores1,2)

            distR1(l)=real(H(k,j))-valores1(l);
            distI1(l)=imag(H(k,j))-valores1(l);

        end

        [valorR1,indR1]=min(abs(distR1));
        [valorI1,indI1]=min(abs(distI1));

        Hest(k,j)=valores1(indR1)+i*valores1(indI1);

    end

end

w3=Hest';

for j=1:size(SNR,2)

R3(j)=log2(det(I+(SNRnat(j)/N)*((w3*H)*(H'*w3'))*inv(w3*w3')));
end

Rm3=mean(R3,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 5 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nbits2=5;

for j=1:M

    for k=1:N

        valores2=[-alpha:(2*alpha)/((2^Nbits2)/2)-
1):alpha];

        for l=1:size(valores2,2)

            distR2(l)=real(H(k,j))-valores2(l);
            distI2(l)=imag(H(k,j))-valores2(l);

        end

        [valorR2,indR2]=min(abs(distR2));
        [valorI2,indI2]=min(abs(distI2));

        Hest2(k,j)=valores2(indR2)+i*valores2(indI2);

    end

end

w4=Hest2';

for j=1:size(SNR,2)

R4(j)=log2(det(I+(SNRnat(j)/N)*((w4*H)*(H'*w4'))*inv(w4*w4')));
end

Rm4=mean(R4,2);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FILTRO LIMITADO A 10 BITS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nbits3=10;

for j=1:M
    for k=1:N
        valores3=[-alpha:(2*alpha)/((2^Nbits3)/2)-
1):alpha];

        for l=1:size(valores3,2)
            distR3(l)=real(H(k,j))-valores3(l);
            distI3(l)=imag(H(k,j))-valores3(l);

        end

        [valorR3,indR3]=min(abs(distR3));
        [valorI3,indI3]=min(abs(distI3));

        Hest3(k,j)=valores3(indR3)+i*valores3(indI3);

    end

end

w5=Hest3';

for j=1:size(SNR,2)

R5(j)=log2(det(I+(SNRnat(j)/N)*((w5'*H)*(H'*w5'))*inv(w5*w5'))));
end

Rm5=mean(R5,2);

end

figure
plot(SNR,R1,'-ro',SNR,R2,'-.b',SNR,R3,'-+g',SNR,R4,'->k',SNR,R5,'-
*m')
title('ESCENARIO 40 ANTENAS TRANSMISORAS Y 64 RECEPTORAS')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')
legend('Matched filter','Matched filter error','3 bit sample
filter','5 bit sample filter','10 bit sample filter')

end

```

MatchedFilterErrorCompare.m

% Función que calcula el filtro adaptado con error, pero en este caso,
% fijando la SNR y variando el error que introducimos.

```
function [R] = MatchedFilterErrorCompare()

    close all
    clear all
    clc

    for i=1:500

        H=canalH();
        M=size(H,2);
        N=size(H,1);

        I=eye(N);

        SNR=10;
        SNRnat=10.^(SNR/10);

        sigma=[0:0.1:1];

        for j=1:size(sigma,2)
            He=H+(sigma(j)*(randn(N,M)+i*randn(N,M)));
            w=He';
            w=w./sqrt(trace(w*w')/M);
            R(j,i)=log2(det(I+((SNRnat/M)*H*(w*w')*H')));
        end

    end

    Rm=mean(R,2);

    figure, stem(sigma,Rm);
    title('CAPACIDAD DEL CANAL CON FILTRO ADAPTADO Y ERROR')
    xlabel('sigma')
    ylabel('Capacidad (bps/Hz)')

end
```

SampleFilter10bitCompare.m

% Función que calcula el filtro limitado a 10 bits, pero en este caso, comparamos como varía la precisión en función del número de antenas.

```
function [] = SampleFilter10bitCompare()

close all
clear all
clc

H1=canalH(10,2);
H2=canalH(40,2);
H3=canalH(64,40);

M1=size(H1,2);
N1=size(H1,1);
M2=size(H2,2);
N2=size(H2,1);
M3=size(H3,2);
N3=size(H3,1);

I1=eye(N1);
I2=eye(N2);
I3=eye(N3);

Hest1=zeros(N1,M1);
Hest2=zeros(N2,M2);
Hest3=zeros(N3,M3);

SNR=[0:1:20];
SNRnat=10.^(SNR/10);

alpha=qfuncinv(0.01)*(1/sqrt(2));

Nbits=10;

for j=1:M1

    for k=1:N1

        valores1=[-alpha:(2*alpha)/(((2^Nbits)/2)-1):alpha];

        for l=1:size(valores1,2)

            distR1(l)=real(H1(k,j))-valores1(l);
            distI1(l)=imag(H1(k,j))-valores1(l);

        end

        [valorR1,indR1]=min(abs(distR1));
        [valorI1,indI1]=min(abs(distI1));

        Hest1(k,j)=valores1(indR1)+1i*valores1(indI1);

    end

end
```

```

end

save 'H1.mat' H1
save 'Hest1t.mat' Hest1

w1=Hest1';
w1=w1./sqrt(trace(w1*w1')/M1);

wa1=H1';
wa1=wa1./sqrt(trace(wa1*wa1')/M1);

for j=1:size(SNR,2)
    R1(j)=log2(det(I1+((SNRnat(j)/M1)*H1*(w1*w1')*H1')));
    R1a(j)=log2(det(I1+((SNRnat(j)/M1)*H1*(wa1*wa1')*H1')));
end

for j=1:M2

    for k=1:N2

        valores2=[-alpha:(2*alpha)/(((2^Nbits)/2)-1):alpha];

        for l=1:size(valores2,2)

            distR2(l)=real(H2(k,j))-valores2(l);
            distI2(l)=imag(H2(k,j))-valores2(l);

        end

        [valorR2,indR2]=min(abs(distR2));
        [valorI2,indI2]=min(abs(distI2));

        Hest2(k,j)=valores2(indR2)+i*valores2(indI2);

    end

end

save 'H2.mat' H2
save 'Hes2t.mat' Hest2

w2=Hest2';
w2=w2./sqrt(trace(w2*w2')/M2);

wa2=H2';
wa2=wa2./sqrt(trace(wa2*wa2')/M2);

for j=1:size(SNR,2)
    R2(j)=log2(det(I2+((SNRnat(j)/M2)*H2*(w2*w2')*H2')));
    R2a(j)=log2(det(I2+((SNRnat(j)/M2)*H2*(wa2*wa2')*H2')));
end

for j=1:M3

    for k=1:N3

        valores3=[-alpha:(2*alpha)/(((2^Nbits)/2)-1):alpha];

```

```

for l=1:size(valores3,2)

    distR3(l)=real(H3(k,j))-valores3(l);
    distI3(l)=imag(H3(k,j))-valores3(l);

end

[valorR3,indR3]=min(abs(distR3));
[valorI3,indI3]=min(abs(distI3));

Hest3(k,j)=valores3(indR3)+i*valores3(indI3);

end

end

save 'H3.mat' H3
save 'Hes3t.mat' Hest3

w3=Hest3';
w3=w3./sqrt(trace(w3*w3')/M3);

wa3=H3';
wa3=wa3./sqrt(trace(wa3*wa3')/M3);

for j=1:size(SNR,2)
    R3(j)=log2(det(I3+((SNRnat(j)/M3)*H3*(w3*w3')*H3')));
    R3a(j)=log2(det(I3+((SNRnat(j)/M3)*H3*(wa3*wa3')*H3')));
end

figure
plot(SNR,R1,'-ro',SNR,R1a,'-r',SNR,R2,'-*b',SNR,R2a,'-
b',SNR,R3,'-+g',SNR,R3a,'-g')
xlabel('SNR (dB)')
ylabel('Capacidad (bps/Hz)')
legend('10 transmisoras, 2 receptoras con filtro estimado 10
bits','10 transmisoras, 2 receptoras con filtro adaptado','40
transmisoras, 2 receptoras con filtro estimado 10 bits','40
transmisoras, 2 receptoras con filtro adaptado','64 transmisoras, 40
receptoras con filtro estimado 10 bits','64 transmisoras, 40
receptoras con filtro adaptado')

end

```


Bibliografía

- [1] Vook, F. W., Ghosh, A., & Thomas, T. A. (2014, June). MIMO and beamforming solutions for 5G technology. In *Microwave Symposium (IMS), 2014 IEEE MTT-S International* (pp. 1-4). IEEE.
- [2] Sánchez Fernández, M., Rajo Iglesias, E., & Quevedo Teruel, O. (2008). *Spectral Efficiency in MIMO Systems Using Space and Pattern Diversities Under Compactness Constraints*. IEEE transactions on vehicular technology.
- [3] Goldsmith, A. (2005). *Wireless communications*. Cambridge University Press.
- [4] Phule, S., & Kazi, A. (2012). *MIMO antenna system for LTE (4G)*. +IEEE Fict International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN).
- [5] Eldin, S. S., Nasr, M., Khamees, S., Sourour, E., & Elbanna, M. (2009, April). LDPC coded MIMO OFDM-based IEEE 802.11 n wireless LAN. In *Wireless and Optical Communications Networks, 2009. WOCN'09. IFIP International Conference on* (pp. 1-5). IEEE.
- [6] Wang, X., Zhou, X., Shen, W., Zou, Z., & Zheng, L. (2014, September). A MIMO-based backscattering RFID with interleave division multiple access for real-time sensing applications. In *RFID Technology and Applications Conference (RFID-TA), 2014 IEEE* (pp. 312-317). IEEE.
- [7] Larson, E., Edfors, O., Tufvesson, F., & Marzetta, T. (2014). *Massive MIMO for next generation wireless systems* (pp. 186-195). IEEE Communications Magazine.
- [8] Romero Ibáñez, I. (2014). *Ortogonalidad y capacidad en sistemas MIMO masivo* (PFC). Universidad Carlos III de Madrid.
- [9] Baker, K. (2013). *Singular Value Decomposition Tutorial*.